

## **2214 - Protecció de la privacitat en entorns urbans i no urbans**

Memòria del projecte Fi de Carrera  
d'Enginyeria en Informàtica  
realitzat per  
*Maria del Camp Davesa Puigvert*  
i dirigit per  
*Xavier Baró Solé i Jordi Vitrià Marca*  
Bellaterra, 22 de Juny de 2010



El sotasignat, *Xavier Baró Solé*

Professor/a de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per na

*Maria del Camp Davesa Puigvert*

I per tal que consti firma la present.

Signat: .....

Bellaterra, 22 de Juny de 2010

# Agraïments

M'agradaria donar les gràcies als meus amics i companys de feina que m'han ajudat a seguir endavant i a fer que aquest projecte arribés a bon port.

Gràcies a la família pel suport donat.

I finalment, donar gràcies al meu director de projecte, en Xavier Baró, que ha tingut molta paciència amb mi i m'ha ajudat en tots els problemes en què m'he trobat al llarg del projecte.



# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
1.1	Objectius . . . . .	3
1.2	Estat de l'Art . . . . .	3
1.3	Anàlisi del problema . . . . .	4
1.3.1	Geomòbil . . . . .	4
1.3.2	Anàlisi de les imatges . . . . .	5
1.3.3	Complexitat dels entorns . . . . .	7
1.3.4	Anàlisi dels mètodes . . . . .	7
1.3.5	Sistema . . . . .	9
1.4	Requeriments . . . . .	9
1.5	Planificació . . . . .	9
<b>2</b>	<b>Metodologia</b>	<b>13</b>
2.1	Detecció de cares . . . . .	13
2.2	Detecció de Matrícules . . . . .	16
2.3	Detecció de Persones . . . . .	19
2.3.1	Caracterstiques de HoG . . . . .	19
2.4	Inpainting . . . . .	21
2.4.1	Imatges Semànticament Iguals . . . . .	22
<b>3</b>	<b>Arquitectura de l'Aplicació</b>	<b>25</b>
3.1	Mòdul de procés . . . . .	26
3.1.1	Tasques . . . . .	26
3.1.2	Treballadors (Workers) . . . . .	29
3.2	Administració del sistema . . . . .	31
3.3	Base de dades . . . . .	33
<b>4</b>	<b>Resultats</b>	<b>35</b>
4.1	Detecció de cares . . . . .	36

4.2	Detecció de persones . . . . .	37
4.3	Detecció de matrícules . . . . .	39
4.4	Inpainting . . . . .	39
<b>5</b>	<b>Conclusions i treballs futurs</b>	<b>43</b>
5.1	Conclusions . . . . .	43
5.2	Treballs futurs . . . . .	44

# Índex de figures

1.1	Exemple d'imatges de diferents bases de dades disponibles. . . . .	2
1.2	La imatge de l'esquerra representa una possible imatge d'entrada del sistema, i la de la dreta seria la sortida d'aquest. . . . .	3
1.3	Furgoneta encarregada de capturar les imatges i algunes imatges d'exemple	5
1.4	A l'esquerra es mostra una cara que ja no es considera identificable, mentre que a la dreta es mostren cares fàcilment identificables. Es mostra la mida en píxels de les cares. . . . .	6
1.5	D'esquerra a dreta es va reduint progressivament la mida en píxels d'un vianant, baixant la resolució de la imatge. A partir d'una alçada d'uns 120 píxels ja comença a ser impossible reconèixer a la persona.. . . .	7
1.6	Mida de matrícules en diferents escenes. . . . .	8
1.7	D'esquerra a dreta es va reduint progressivament la mida en píxels d'una matrícula, baixant la resolució de la imatge. A partir d'una alçada màxima d'uns 11 píxels ja comença a ser impossible llegir els dígit. . . . .	8
1.8	Esquema del sistema implementat. . . . .	9
1.9	Planificació del projecte. A (a) hi ha representada la planificació inicial, i a (b) la final. . . . .	11
2.1	Procés de detecció mitjançant finestra lliscant. . . . .	13
2.2	Cascada de detectors. Cada nivell consisteix en un detector robust, format per una combinació de detectors senzills apresats utilitzant AdaBoost. Les regions que no contenen cares es van descartant nivell a nivell, mentre que les que contenen una cara arriben al final de la cascada. . . . .	14
2.3	Tipus de filtres de Haar. El valor d'aquestes característiques s'obté restant la suma de tots els punts dins de les regions negres, de la suma de tots els punts en les regions blanques. D'esquerra a dreta tenim els detectors de contorns, línies horitzontals i verticals, regions aniuades i finalment línies diagonals. . . . .	16

2.4	A l'esquerra podem veure la definició d'imatge integral (II), on el punt $(x, y)$ conté el valor de la suma de la regió fosca. A la part inferior es mostra la fórmula que utilitzarem per trobar la suma de tots els punts d'una regió qualsevol com la que es mostra a la part superior dreta. . . . .	16
2.5	Representació d'una imatge utilitzant el gradient. a) imatge de prova. S'han creat dues regions on els colors estan invertits. b) Mòdul del gradient. c) Mòdul del gradient en sentit horitzontal, d) Mòdul del gradient en sentit vertical. . . . .	17
2.6	D'esquerra a dreta, mitja dels valors del gradient en horitzontal, vertical i del mòdul. . . . .	18
2.7	Patrons definits per a les característiques rectangulars. . . . .	18
2.8	A l'esquerra tenim el resultat d'analitzar una imatge amb la cascada de detectors. A la dreta es mostren les regions després del procés d'agupació, indicant el nombre de subregions que les formen. . . . .	19
2.9	Esquema del funcionament de HOG. . . . .	20
2.10	Exemple de les imatges obtingudes a l'analitzar la base de dades que fa de biblioteca per obtenir imatges semblants a l'original. . . . .	23
2.11	Exemple de les imatges obtingudes per la Geomòbil. La imatge central representa la imatge d'entrada i les altres quatre, les que s'utilitzarien per portar a terme la reconstrucció. . . . .	23
3.1	Esquema de blocs de l'aplicació. . . . .	26
3.2	Diagrama de la vida de l'aplicació. . . . .	28
3.3	Exemple de la execució d'un pla. . . . .	29
3.4	Vida d'un worker. . . . .	30
3.5	Informació mostrada pel mòdul de seguiment de tasques i Workers. . . . .	32
3.6	Esquema de la base de dades. . . . .	34
4.1	Exemples de detecció de cares. (a) i (b) Deteccions de cares bones. (c) Imatge cremada amb un fals negatiu. (d) Cara amb complements. (e) Cara amb un angle molt tancat. (f) Mala detecció, en aquest cas degut a que la reixa té formes similars a una cara. . . . .	37
4.2	Exemples de detecció de persones. (a) Detecció correcta de les persones. (b) Mala detecció de les persones. . . . .	38
4.3	Exemples de detecció de matrícules. (a) Correcta detecció d'una matrícula, també es veu com en matrícules antigues, al ser una regió poc ampla, no es detecta. (b) Exemple d'un error en la detecció de matrícules. (c) Exemple d'un error en la detecció de matrícules. (d) Regió corresponent a la figura 4.3c, on s'ha eliminat la rotació. En aquest cas les matrícules si que es poden detectar. . . . .	40



4.4	Procés del sistema amb aplicació de l’Inpainting . . . . .	41
4.5	Un exemple de resultats dolents utilitzant l’Inpainting. . . . .	41



# Capítol 1

## Introducció

En els últims anys hi ha hagut un notable increment en el nombre d'imatges urbanes que es poden trobar a la web. Tot i que la més àmplia i coneguda base de dades d'imatges és la de Google Street View<sup>1</sup>, n'hi ha moltes d'altres de més petites, com poden ser Mapjack<sup>2</sup>, Everyscape<sup>3</sup> o Daum's Road View<sup>4</sup>, un exemple d'aquestes imatges el podem veure a la figura 1.1. El nostre sistema es basa en imatges capturades per la Geomòbil [1], un sistema de cartografia mòbil desenvolupat per l'Institut Cartogràfic de Catalunya (ICC)<sup>5</sup>. Aquest sistema consisteix en una furgoneta amb sis càmeres calibrades i sincronitzades amb un sistema GPS/INS que permet capturar i georeferenciar diferents imatges simultàniament. Utilitzant la Geomòbil, més de 10 km. de carrers de Barcelona han estat capturats i georeferenciats, el que correspon a més de 50.000 imatges.

Aquest conjunt d'imatges representa una gran font d'informació per poder dur a terme diferents tasques municipals, com pot ser el cas de portar un control sobre l'estat de les façanes dels edificis o la localització de diferents establiments, entre d'altres. Aquesta informació també pot ser utilitzada per navegar per la ciutat o per la comunitat de recerca en visió per computador. No obstant això, com que les imatges han estat capturades en condicions reals, moltes d'elles contenen gent caminant, cotxes i altre tipus d'informació sensible que cal tenir en compte abans de publicar-les, ja que s'ha de garantir la privacitat de les persones.

Per garantir aquesta privacitat cal eliminar totes les cares i matrícules de les imatges, tal i com fa Google [2] en la seva base d'imatges a disposició del públic. Però ens podem trobar que a vegades, amb el sol fet d'eliminar la cara, la persona encara sigui reconeix-

---

<sup>1</sup>[maps.google.com/help/maps/streetview](http://maps.google.com/help/maps/streetview)

<sup>2</sup>[www.mapjack.com](http://www.mapjack.com)

<sup>3</sup>[www.everyscape.com](http://www.everyscape.com)

<sup>4</sup>[local.daum.net/map](http://local.daum.net/map)

<sup>5</sup>[www.icc.cat](http://www.icc.cat)



**Figura 1.1:** Exemple d'imatges de diferents bases de dades disponibles.

ible, ja sigui per l'entorn o per la manera de vestir o per altres factors, i per tant, no s'ha eliminat tota la informació sensible de la imatge. És per això que en aquest projecte es proposa un sistema de protecció de la privacitat on també s'eliminin les persones de la imatge.

Aquest sistema forma part d'un projecte ja creat. L'inici del projecte parteix de la necessitat d'analitzar de forma automàtica el gran volum de dades generat per la Geomòbil. Ja d'entrada es classifica aquest problema dins del camp del reconeixement d'objecte i es comença la recerca partint d'estudis ja realitzats en aquest camp per resoldre problemes similars aïllant aquelles metodologies amb resultats més prometedors i aplicables en aquest cas.

Després de la implementació i prova de diferents vies, s'arriba al disseny d'un primer sistema de reconeixement de senyals, que deriva en un primer prototip. Aquest prototip combina diferents metodologies per a la detecció i posterior classificació dels senyals de trànsit, amb heurístiques que milloren els resultats utilitzant els trets diferencials del problema.

## 1.1 Objectius

L'objectiu del projecte és analitzar el problema de la privacitat en les imatges de la Geomòbil i desenvolupar un software que ens permeti eliminar la informació sensible continguda en aquestes. Donada la durada limitada del projecte, s'estudiaran els mètodes existents per a localitzar cares, persones i matrícules. Un cop localitzades les regions amb contingut sensible, s'estudiaran mètodes per eliminar aquestes regions de manera que afectin el mínim possible a la imatge final, és a dir, que el resultat final sigui coherent a la vista.

A la figura 1.2 es pot veure una imatge i com quedaria un cop processada pel sistema, és a dir, que les cares i les matrícules quedarien difuminades, i les persones eliminades sense alterar la imatge.



**Figura 1.2:** La imatge de l'esquerra representa una possible imatge d'entrada del sistema, i la de la dreta seria la sortida d'aquest.

## 1.2 Estat de l'Art

A la literatura trobem molts treballs que tracten la protecció de la privacitat de les persones en imatges, però cal destacar el treball realitzat a Google Street View [2], ja que actualment és la base d'imatges a disposició del públic més utilitzada. En aquesta aplicació, les cares s'aconsegueixen eliminar amb una precisió del 89.0%, i les matrícules amb una precisió del 96.5% les d'Estats Units i un 93.6% les europees.

Un sistema comú per protegir la privacitat de les persones a les imatges, és la utilització d'un software detector de cares que és utilitzat per difuminar aquestes cares i fer que les persones no siguin reconegudes. Les variacions de l'aparença que poden tenir les cares, la il·luminació, les expressions i altres factors transformen el procés de detecció de

cares en un problema complex de reconeixement de patrons, on les tècniques basades en l'aprenentatge constitueixen l'estratègia més efectiva per construir classificadors cara/no-cara [3]. Un dels mètodes més coneguts per a la detecció de cares és el proposat per Viola i Jones a [4].

Pel que fa a la literatura referent als vianants, es troba molta informació en el cas de la detecció de vianants orientada a l'assistència en conducció [5], és a dir, a ajudar al conductor a evitar accidents amb vianants o en fer que aquests siguin el més lleu possible, però encara no s'ha utilitzat aquesta tècnica per a la protecció de la privacitat en imatges. Un mètode molt estès per a la detecció de persones és mètode de HOG [6], on s'utilitza un histograma de gradients orientats per a realitzar la detecció.

A la literatura trobem molts treballs que tracten la detecció de regions de text dins de documents, però molt pocs que afrontin el problema en entorns no controlats com poden ser imatges de carretera o entorns urbans. En aquest sentit cal destacar el treball realitzat a [7] on es detecten regions de text en imatges urbanes.

## 1.3 Anàlisi del problema

L'objectiu del projecte és trobar les regions amb contingut sensible d'una imatge. Les imatges que ens interessa processar són les obtingudes pel projecte Geomòbil, per tant, hem de considerar que es troben en un entorn no controlat, el que vol dir que no tenim cap control sobre la il·luminació ni l'entorn en què es troben aquestes regions.

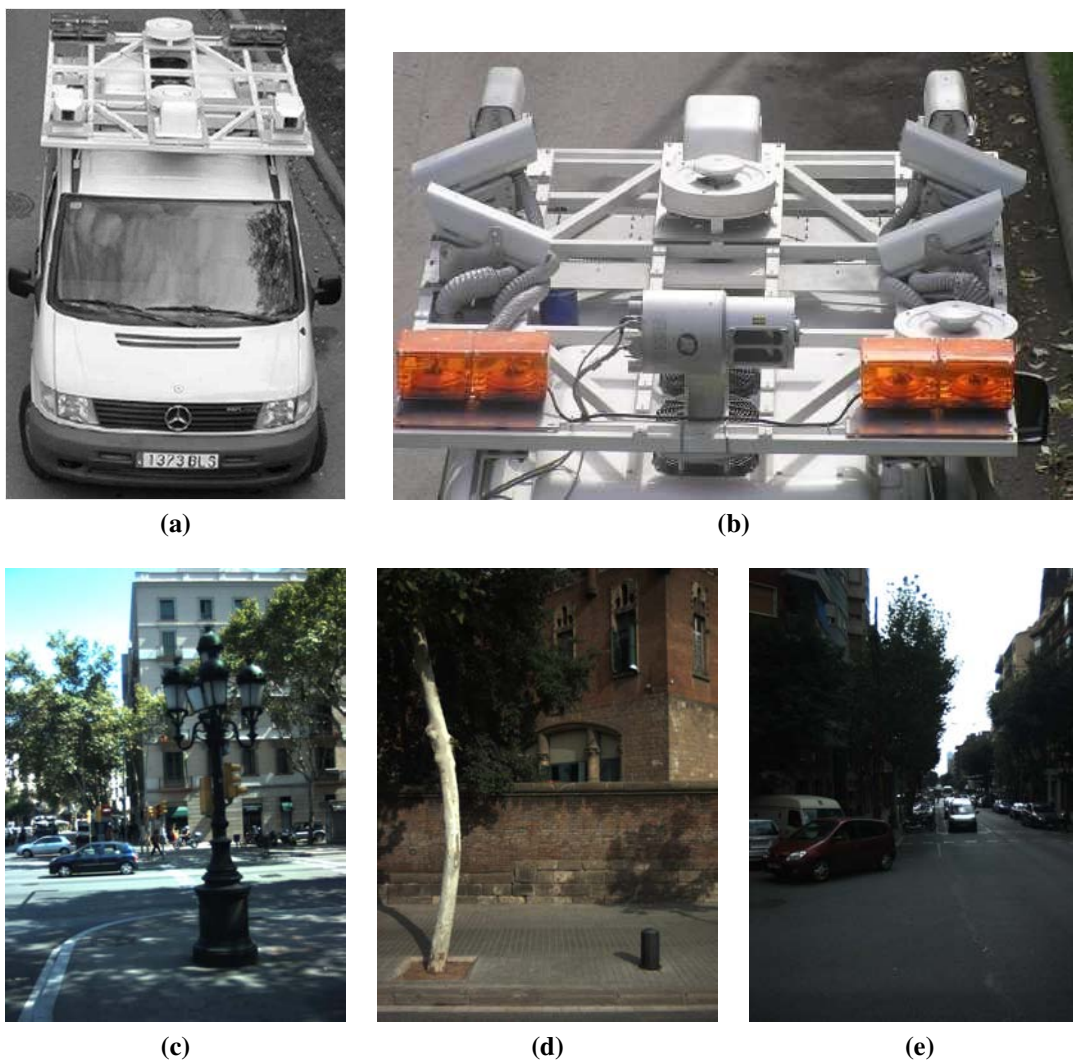
### 1.3.1 Geomòbil

El sistema de cartografia mòbil [8] utilitzat en aquest projecte s'ha desenvolupat per tal de crear un inventari dels carrers de Barcelona, especialment per capturar les façanes dels edificis per comprovar-ne el seu estat. Un total de més de 10km. de carrers han estat explorats i capturats utilitzant un sistema de sis càmeres en color amb diferents orientacions, totes elles sincronitzades amb un sistema GPS/INS, adquirint una imatge cada cinc metres aproximadament. El resultat és un conjunt d'unes 50.000 imatges georreferenciades. El vehicle utilitzat per a la captura d'imatges i alguns exemples de captures de la ciutat de Barcelona es poden veure a la figura 1.3.

Les imatges són emmagatzemades utilitzant el format *LizardTech MrSid*<sup>6</sup> amb una resolució de  $1200 \times 1600$  píxels.

---

<sup>6</sup>[www.lizardtech.com](http://www.lizardtech.com)



**Figura 1.3:** Furgoneta encarregada de capturar les imatges i algunes imatges d'exemple

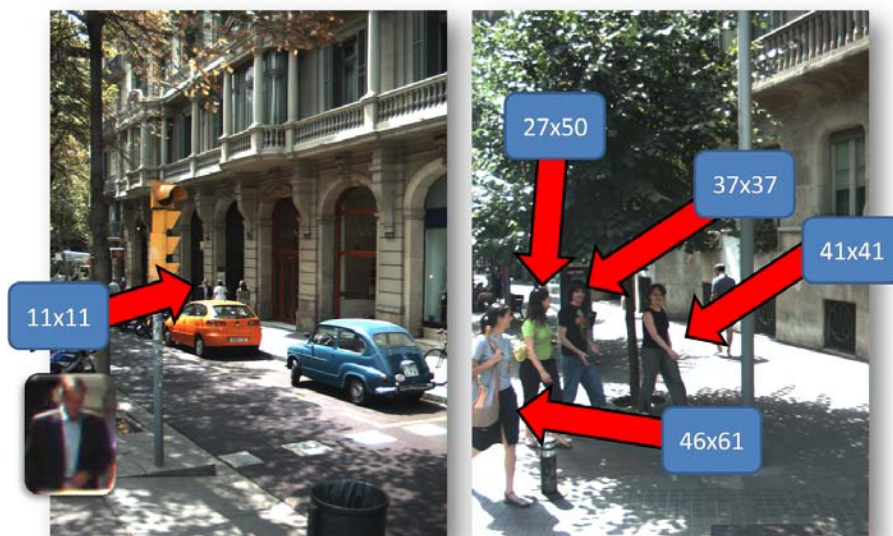
### 1.3.2 Anàlisi de les imatges

Un cop s'han obtingut les imatges, cal mirar com apareixen les persones i els cotxes en aquestes i decidir a partir de quina mida creiem que la persona i la matrícula són reconeixibles, ja que la mida de les cares, dels vianants i de les matrícules difereix molt segons la distància a què es troben de la càmera i, a més a més, moltes de les cares apareixen de perfil.

A la figura 1.4 es mostren dues escenes on apareixen persones a diferents distàncies i



s'hi mostra la mida de les cares.



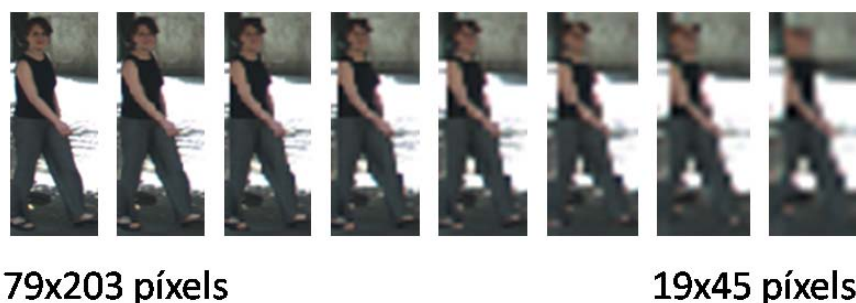
**Figura 1.4:** A l'esquerra es mostra una cara que ja no es considera identificable, mentre que a la dreta es mostren cares fàcilment identificables. Es mostra la mida en píxels de les cares.

Cal tenir en compte que tot i que en algun cas la cara és molt petita, el fet de que aparegui la persona sencera fa que tot i no ser fàcil reconèixer la persona mitjançant la cara, si que amb la informació de la resta del cos podem arribar a identificar de quina persona es tracta. A la figura 1.5 es mostra un vianant canviant la seva resolució progressivament per fixar una mida a partir de la qual es consideri que ja no es pot reconèixer aquesta persona. Tenint en compte que la cara és el 10% de l'alçada de la persona, hem fixat que la mida a partir de la qual podem reconèixer una persona és de  $12 \times 12$  píxels, mentre que si només tenim la cara, aquesta mida ha de ser més gran, sobre els  $18 \times 18$  píxels.

De la mateixa manera que hem fet en el cas de les cares, cal identificar a partir de quina mida som capaços de llegir el contingut d'una matrícula i, per tant, a partir de quina mida és interessant detectar-les. A la figura 1.6 es mostren dues escenes amb matrícules, i s'hi indica la seva mida en píxels.

Per determinar la mida mínima s'ha agafar una matrícula i se li ha anat disminuint la seva resolució progressivament (veure figura 1.7). S'ha determinat que per sota dels 11 píxels d'alçada, els dígit no poden ser reconeguts.





**Figura 1.5:** D'esquerra a dreta es va reduint progressivament la mida en píxels d'un vianant, baixant la resolució de la imatge. A partir d'una alçada d'uns 120 píxels ja comença a ser impossible reconèixer a la persona..

### 1.3.3 Complexitat dels entorns

Un cop descrit el problema, ara analitzarem la complexitat dels diferents entorns en els que opera la Geomòbil.

- **Carretera:** Les regions d'interès en carretera seran les matrícules dels cotxes, que en aquest cas apareixeran sobretot frontalment de manera que es llegeixen fàcilment. A part de les regions amb matrícules, altres regions que també poden aparèixer amb més freqüència, són les que contenen cares, que seran les cares dels diferents components dels cotxes de la carretera.
- **Entorn urbà:** Les regions d'interès són molt variables, ja que en aquest cas és quan es poden detectar més vianants. A més a més, en aquest cas, moltes matrícules apareixen esbiaixades, ja que no sempre es té el cotxe a davant, sinó que estan aparcats, o coincideixen en un xamfrà.

### 1.3.4 Anàlisi dels mètodes

Per poder solucionar el problema de la protecció de la privacitat de les persones en les imatges, cal primer analitzar els mètodes que hi ha disponibles.

- **Detector de cares:** Un problema que ens trobem és el de la detecció de cares, ja que les cares poden tenir barba, bigoti, ulleres o portar altres complements com per exemple gorres que facin difícil la seva detecció. Fins al moment, el mètode més utilitzat és del de Viola i Jones.
- **Detector de matrícules:** Les matrícules consten de números i lletres, i per tant, segueixen uns patrons determinats. El principal problema en què ens trobem és



**Figura 1.6:** Mida de matrícules en diferents escenes.



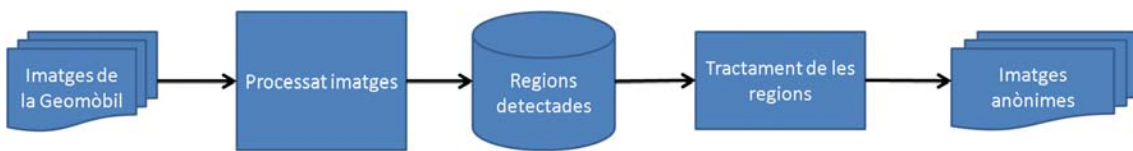
**Figura 1.7:** D'esquerra a dreta es va reduint progressivament la mida en píxels d'una matrícula, baixant la resolució de la imatge. A partir d'una alçada màxima d'uns 11 píxels ja comença a ser impossible llegir els dígit.

en les rotacions, ja que no sempre les matrícules apareixen de cara. Per solucionar aquest tema, s'utilitzarà un detector de text implementat anteriorment en el sistema, però adaptat per a què només detecti les matrícules.

Però creiem que només amb la utilització d'aquests dos mètodes no n'hi ha prou, ja que tal i com s'ha dit anteriorment, encara que es difumini la cara d'una persona, aquesta pot continuar essent reconeguda, ja sigui per l'entorn a on està, per la manera de vestir o per altres complements. És per això que s'ha decidit implementar un altre mètode, el detector de persones. Però a diferència de la detecció de cares, en lloc de difuminar a la persona completa, cosa que faria distorsionar la imatge original, el que es farà serà eliminar-la de la imatge, és per aquest motiu, que serà necessari implementar l'Inpainting per poder reconstruir la imatge un cop la regió s'hagi eliminat.

### 1.3.5 Sistema

A la figura 1.8 hi ha un esquema del funcionament del sistema implementat. Al sistema se li entren les imatges a processar, en aquest cas les procedents de la Geomòbil, a continuació de detecten totes les regions amb contingut sensible que posteriorment seran tractades. Les corresponents a cares i matrícules seran difuminades, i les corresponents a vianants, seran eliminades. Un cop tractades les regions, s'aplicarà l'Inpainting a totes aquelles imatges on s'hagin detectat vianants per poder-les reconstruir.



**Figura 1.8:** Esquema del sistema implementat.

## 1.4 Requeriments

Al ser un sistema que tracta amb un gran volum de dades i d'imatges en el qual es poden detectar múltiples regions per imatges i fer-ne la posterior reconstrucció, no es demana que sigui un sistema a temps real. La part referent als detectors, tant de persones com de cares i matrícules, estan implementats en C++ utilitzant les llibreries OpenCV 2.0, i la part de l'Inpainting, al ser la part més experimental i en fase d'investigació, s'ha implementat en Matlab. Per poder aprofitar la versatilitat del llenguatge Matlab i la potència i velocitat del C++, s'han programat fitxers en Mex que permeten comunicar els dos llenguatges.

## 1.5 Planificació

El desenvolupament d'aquest projecte s'ha dut a terme amb dues fases diferenciades. El propòsit general per la primera fase era comprovar la viabilitat del projecte, que consistia en adquirir informació sobre el tema de la privacitat de les imatges en altres bases de dades i la primera implementació d'alguns mètodes. El propòsit general per la segona fase era la consolidació dels mètodes implementats a la primera fase i buscar informació sobre la detecció de matrícules i la seva implementació. En les dues fases es contempla l'escriptura del projecte.

- **Primera fase**

- **Estudi viabilitat:** Aquesta part ha consistit en la recerca d'informació sobre la protecció de la privacitat de les persones en imatges a disposició del públic. També s'ha mirat que fos viable implementar aquest sistema en l'estructura creada fins al moment i quins mètodes són els que s'hi adaptarien millor.
- **Detecció de cares:** Un cop s'ha escollit el mètode més adequat, s'ha procedit a la seva implementació.
- **Detecció de persones:** És el mateix cas que el detector de cares. Un cop escollit el mètode que es considera més adient, s'ha procedit a la seva implementació.
- **Anàlisi de Resultats:** S'han analitzat els resultats obtinguts amb aquests dos mètodes per valorar si eren prou bons com per continuar amb la mateixa línia de desenvolupament o calia un canvi de mètodes i de punt de vista.

#### • Segona fase

- **Estudi del detector de matrícules:** Un cop s'ha vist que el projecte era viable, s'ha procedit a buscar informació sobre la detecció de matrícules per a la protecció de la privacitat i escollir el millor en el nostre cas.
- **Detecció de matrícules:** En el mateix cas que els anteriors, un cop triat el mètode a utilitzar, s'ha implementat integrant-lo en el sistema.
- **Prova i anàlisi de tots els mètodes:** Un cop implementats tots els mètodes, s'ha aplicat conjuntament a les imatges per poder obtenir resultats finals i decidir si el sistema implementat és bo o no.

A part d'aquestes fases, també s'ha anat escrivint la memòria del projecte, d'aquesta manera, cada part anava quedant documentada.

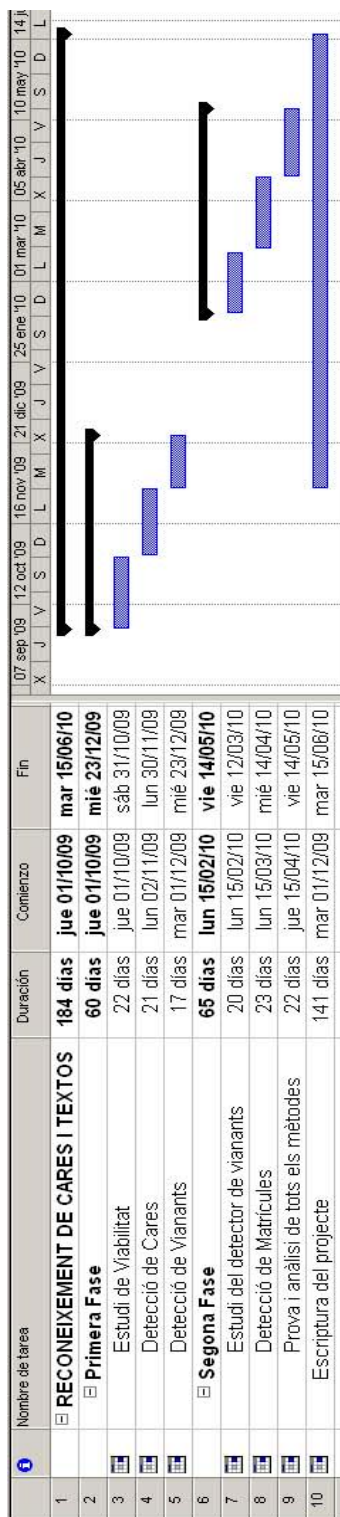
Fins aquí s'ha descrit la planificació inicial del projecte, tal i com es pot veure a 1.9a.

La planificació final ha variat força respecte a la inicial. La principal diferència ha estat que el detector de matrícules s'ha pogut analitzar i implementar a la primera fase del projecte. Degut a aquest fet, i com que s'ha vist en els resultats que el fet de difuminar a les persones distorsionava molt el resultat final de les imatges, ha estat quan s'ha decidit buscar informació i implementar el mètode de l'Inpainting.

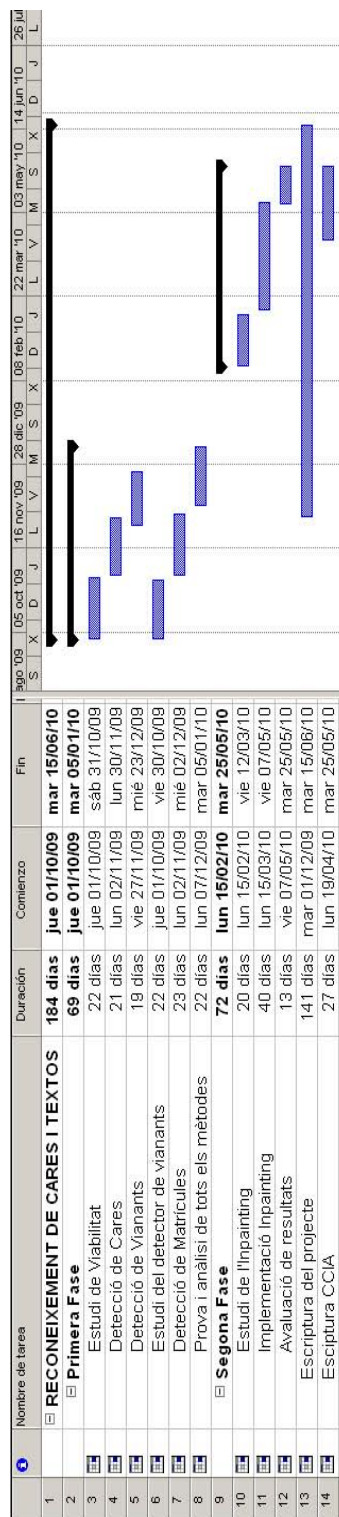
Donat que el projecte estava en fase de finalització quan en el moment de la crida a la participació del CCIA<sup>7</sup>, i donat que el seu contingut encaixava en aquest congrés, es va enviar un resum del projecte en format article. Per aquest motiu s'ha afegit una altra tasca a la planificació, la de l'escriptura de l'article. A la figura 1.9b es pot veure com ha quedat la planificació final.

---

<sup>7</sup><http://deim.urv.cat/ccia2010>



**(a)**



**(b)**

**Figura 1.9:** Planificació del projecte. A (a) hi ha representada la planificació inicial, i a (b) la final.



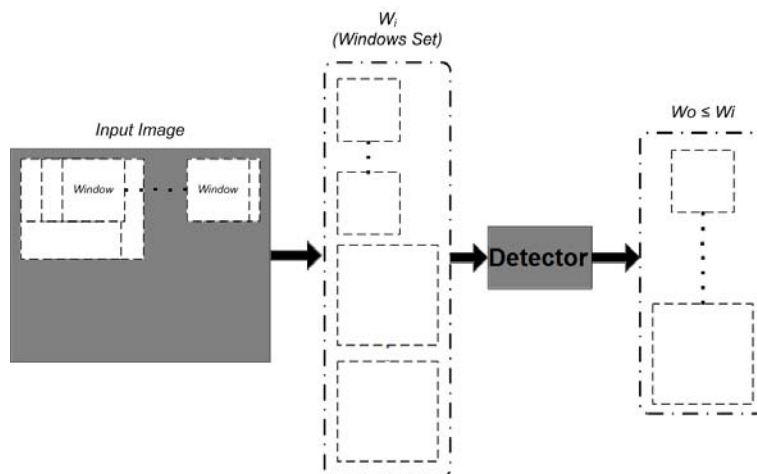
# Capítol 2

## Metodologia

Un cop analitzades les imatges que s'obtenen de la Geomòbil i els possibles problemes que poden derivar d'elles, es procedeix a explicar els mètodes que s'han implementat per dur a terme la protecció de la privacitat en aquestes imatges.

### 2.1 Detecció de cares

La detecció de cares es realitza mitjançant una finestra lliscant sobre la imatge d'entrada, classificant cada regió com a cara o no cara. Aquest sistema està basat en el sistema clàssic proposat per Viola i Jones [4](veure figura 2.1).

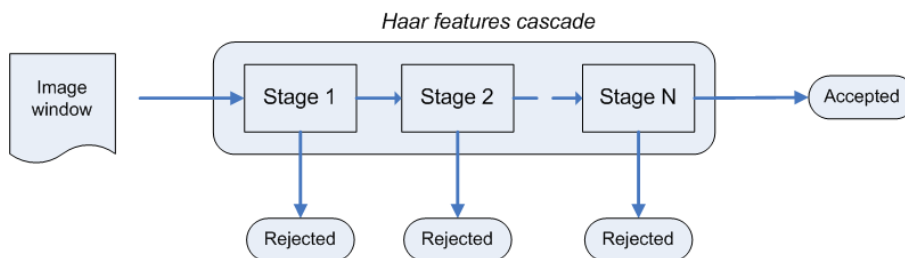


**Figura 2.1:** Procés de detecció mitjançant finestra lliscant.

En l'article de Viola i Jones es presenta tot un esquema d'aprenentatge que consisteix

en la combinació utilitzant Adaboost de classificadors molt senzills basats en filtres Haar. Cada classificador simple es compon d'un filtre i un llindar a partir del qual es decideix si la regió analitzada és una cara o no. El resultat d'aquesta combinació és un detector binari robust, que decideix si una regió és o no una cara.

Donat el gran nombre de regions que cal analitzar per cada imatge d'entrada, s'utilitza un detector en forma de cascada, on cada nivell correspon a un detector binari robust i que sols s'aplica a les regions que han estat classificades com a cara en els classificadors previs a la cascada (veure figura 2.2). Totes aquelles regions que arriben a l'últim nivell i són classificades en aquest nivell com a cares, són considerades regions amb una cara, i per tant, el resultat del procés.



**Figura 2.2:** Cascada de detectors. Cada nivell consisteix en un detector robust, format per una combinació de detectors senzills apresats utilitzant AdaBoost. Les regions que no contenen cares es van descartant nivell a nivell, mentre que les que contenen una cara arriben al final de la cascada.

Tot i que a la literatura es poden trobar diferents procediments per combinar classificadors dèbils, quan s'utilitzen característiques rectangulars per a la detecció de cares, l'algorisme més àmpliament acceptat és l'AdaBoost [9]. A l'algorisme 1 es mostren les versions de l'AdaBoost que utilitzarem per aprendre el classificador. Donat un conjunt d'aprenentatge, aquest algorisme utilitza una distribució de pesos  $W$  sobre els exemples per anar seleccionant classificadors dèbils en funció dels que s'han seleccionat prèviament. El resultat final és una hipòtesis que combina diverses hipòtesis dèbils en una única hipòtesi robusta, la qual ens permet classificar una regió entre cara i no cara.

Les característiques de Haar són un descriptor àmpliament utilitzat en la detecció d'objectes, i especialment en la detecció de cares. Donada una imatge  $\mathcal{I} \in \mathbb{N}_{w \times h}$  entenem per característica tota funció  $\mathfrak{F} : \mathcal{I} \mapsto \mathbb{R}$ . En el cas concret de les característiques de Haar, el valor es calcula comparant els valors de dues o més regions contigues, les quals estan disposades en configuracions especials per a detectar estructures dins de la imatge. Hi ha filtres per detectar contorns, línies i regions aniuades (veure figura 2.3).

Donat que el càlcul dels filtres consisteix en comparacions entre dues o més regions d'una imatge, fa que aquestes característiques siguin robustes davant de canvis d'il·luminació, i al utilitzar regions en comptes de punts aïllats, obtenim robustesa també



---

**Algorithm 1** Gentle Adaboost [9]

---

**Require:**

Seqüència de  $M$  exemples etiquetats  $(x_1, y_1), \dots, (x_M, y_M)$  amb  $y_i \in \{-1, +1\}$

Inicialitzar els pesos  $W_1(i) = 1/M$

**for**  $t = 1, \dots, T$  **do**

    Trobar la hipòtesis dèbil  $h_t(x)$  tal que minimitzi

$$\epsilon_t = Pr_{i \sim W_t}[h_t(x_i) \neq y_i]$$

    Guardar la hipòtesis  $h_t(x) \mapsto \{-1, +1\}$  amb l'error  $\epsilon_t$

    Actualitzar els pesos:

$$W_{t+1}(i) \leftarrow W_t(i) \times \exp(-y_i \times h_t(x_i))$$

    Normalitzar  $W_{t+1}$  de tal manera que

$$\sum_{i=1}^M W_{t+1}(i) = 1$$

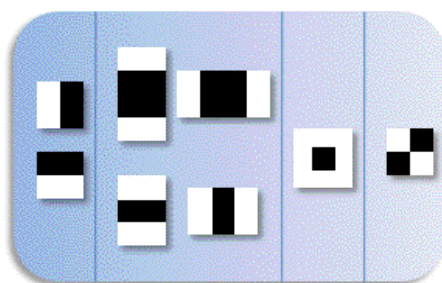
**end for**

**Ensure:** el classificador final:

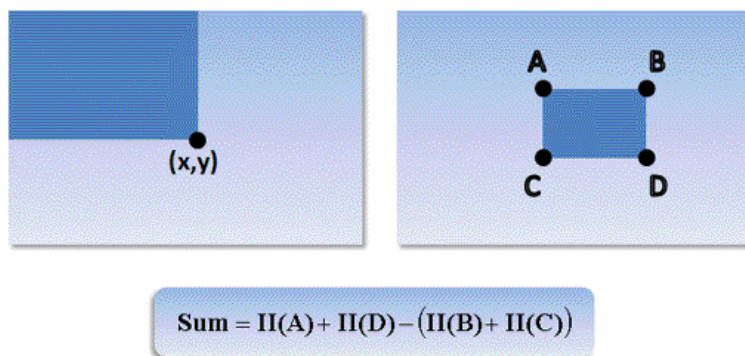
$$H(x) = \text{sign} \left( \sum_{t=1}^T h_t(x) \right)$$


---

davant de soroll. Una altra propietat de les característiques de Haar és que poden ser calculades amb molt poc cost computacional gràcies a la utilització de la imatge integral, la qual consisteix en què el punt  $(x, y)$  conté la suma de tots els punts de la regió marcada. Tal i com es mostra a la figura 2.4, calcular la suma de tots els punts d'una regió es pot fer simplement amb l'accés a 4 posicions de la imatge integral.



**Figura 2.3:** Tipus de filtres de Haar. El valor d'aquestes característiques s'obté restant la suma de tots els punts dins de les regions negres, de la suma de tots els punts en les regions blanques. D'esquerra a dreta tenim els detectors de contorns, línies horitzontals i verticals, regions aniuades i finalment línies diagonals.



**Figura 2.4:** A l'esquerra podem veure la definició d'imatge integral (II), on el punt  $(x, y)$  conté el valor de la suma de la regió fosca. A la part inferior es mostra la fórmula que utilitzarem per trobar la suma de tots els punts d'una regió qualsevol com la que es mostra a la part superior dreta.

## 2.2 Detecció de Matrícules

A diferència del que passava amb les cares, les matrícules sols apareixen quan tenim els cotxes de cara, i això acostuma a passar només en xamfrans o en les càmeres davanteres de la Geomòbil, que en una ciutat tenen una utilitat molt limitada. El gran problema de les matrícules és que apareixen sovint amb rotacions i deformacions.

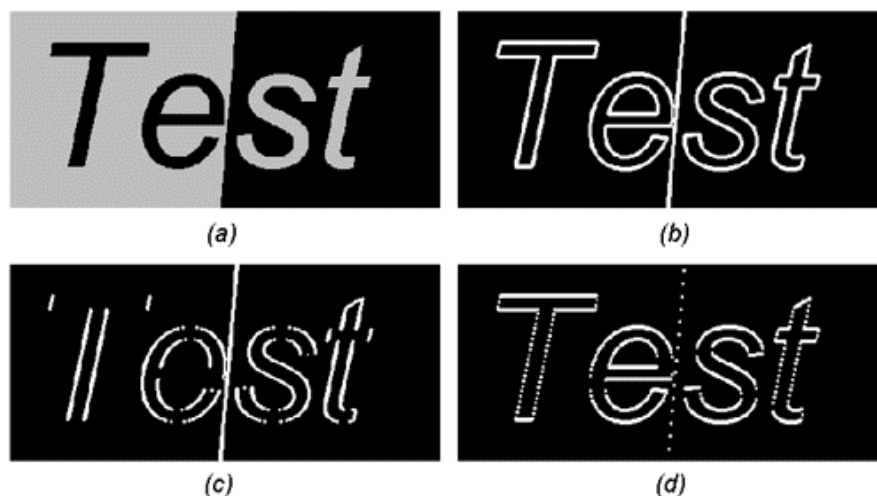
Donat que una matrícula és una regió de text, s'ha utilitzat el detector de regions de text per tal de trobar les matrícules. La gran variabilitat que existeix entre les diferents regions de text, inclús en el cas de les matrícules, on ens podem trobar amb matrícules noves, velles i de diferents països, obliga a buscar descriptors que no siguin molt dependents de l'aspecte de les regions.

L'esquema és molt similar al detector de cares, però les característiques de Haar són

molt dependents de l'aspecte de la regió, per tant, no es poden utilitzar en aquest cas. El que farem és buscar característiques que es fixin amb l'estructura del text en comptes del seu aspecte. Per descartar l'aspecte i extreure l'estructura utilitzarem una representació dels contorns de la imatge original. Crearem tres representacions per a la imatge:

- Mòdul del gradient en direcció vertical
- Mòdul del gradient en direcció horitzontal
- Mòdul del gradient

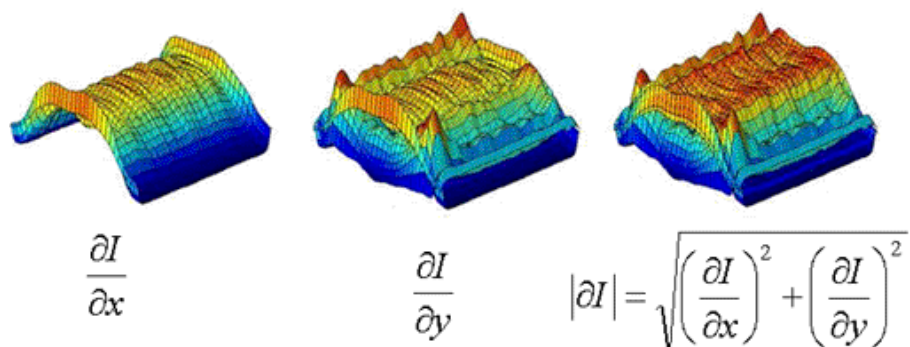
Tal i com es pot veure a la figura 2.5, les tres representacions només capten canvis en el contrast, i per tant, lletres clares sobre fons fosc i lletres fosques sobre fons clar, donen exactament la mateixa resposta.



**Figura 2.5:** Representació d'una imatge utilitzant el gradient. a) imatge de prova. S'han creat dues regions on els colors estan invertits. b) Mòdul del gradient. c) Mòdul del gradient en sentit horitzontal, d) Mòdul del gradient en sentit vertical.

Per veure els trets característics de les regions que contenen text des del punt de vista del seu gradient, el que fem és calcular una mitjana dels valors d'aquestes representacions sobre un conjunt d'imatges. Tal i com es pot veure a la figura 2.6, les imatges de gradient tenen una estructura molt marcada, i per tant, intentarem buscar un conjunt de característiques rectangulars que siguin capaces de captar aquesta estructura tant característica.

El primer que veiem, és que tenim estructures simètriques tant horitzontalment com verticalment. A més a més, a la part superior i a la part inferior de les regions tenim valors



**Figura 2.6:** D'esquerra a dreta, mitja dels valors del gradient en horitzontal, vertical i del mòdul.



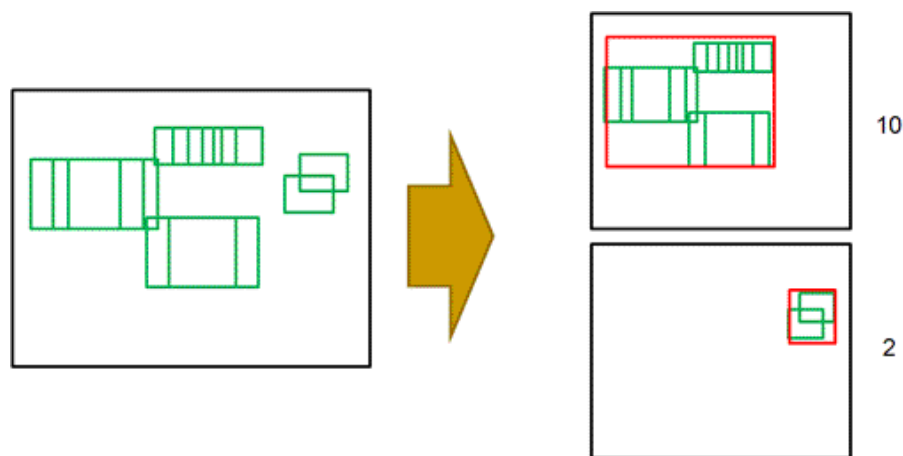
**Figura 2.7:** Patrons definits per a les característiques rectangulars.

molt petits de gradient. Per aprofitar aquests fets, s'han definit els patrons que es mostren a la figura 2.7.

El càlcul del valor d'una característica es fa utilitzant algunes de les regions del patró com a valor positiu, i les restants com a valor negatiu. A més a més, les característiques es poden avaluar sobre qualsevol de les tres imatges de gradient que tenim. En principi es deixa oberta qualsevol possible combinació, i durant l'aprenentatge es triaran aquelles combinacions que millor s'adaptin a les imatges de text. Per calcular el valor de cada regió, utilitzarem la imatge integral calculada a partir de les imatges de gradients, tal com es feia en el cas de les cares.

El procés d'aprenentatge i detecció és igual al cas de les cares, simplement s'han canviat les característiques per unes altres més aptes per a la descripció de regions de text. A diferència de les cares, on obtenim una regió per cara, en el del text s'acostuma a obtenir moltes deteccions per a una mateixa regió de text, ja que l'estructura es repeteix en tota la regió. Per tal d'obtenir les regions de text finals i a la vegada eliminar falsos positius obtinguts pel nostre detector, el que farem és crear grups de regions amb sol·lapament (veure figura 2.8).

Un cop feta l'agrupació, s'eliminen aquelles regions que contenen un nombre molt



**Figura 2.8:** A l'esquerra tenim el resultat d'analitzar una imatge amb la cascada de detectors. A la dreta es mostren les regions després del procés d'agupació, indicant el nombre de subregions que les formen.

petit de deteccions, donant les regions resultants de l'agrupació com a resultat del procés.

## 2.3 Detecció de Persones

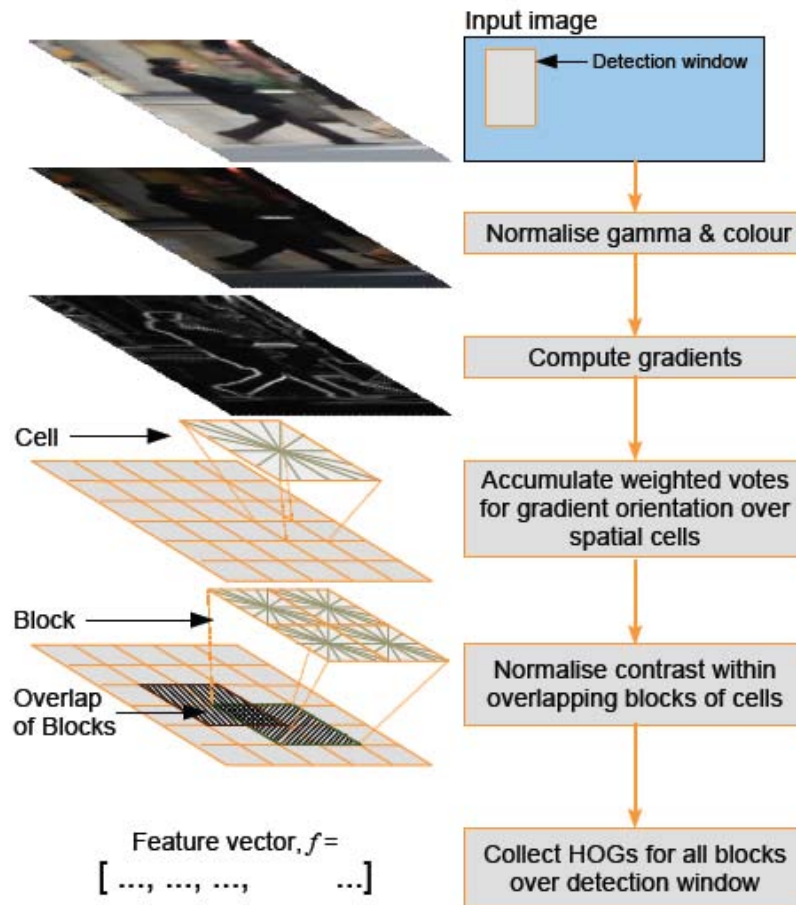
Donat que el detector de cares explicat anteriorment no dóna molts bons resultats en cares molt petites, també s'ha incorporat un detector de persones, que ens permet detectar tota la persona en aquells casos en què la cara sigui massa petita. La detecció de persones en imatges és una tasca difícil, sobretot a causa de que és un objecte deformable, i amb una gran variabilitat en el seu aspecte.

La detecció de persones, s'ha estudiat àmpliament en els camps de la seguretat i l'assistència a la conducció, i la tècnica que sembla donar millors resultats en la literatura és la utilització d'histogrames sobre les orientacions de gradient (Histogram of Oriented Gradients -HOG), introduït per Dalal i Triggs a [6]. De forma similar al cas del detector de cares, la imatge d'entrada s'analitza mitjanant una finestra lliscant, però en aquest cas, la finestra en comptes de ser quadrada, guarda les proporcions del cos humà. En aquest cas, en comptes d'utilitzar els filtres de Haar per a descriure les imatges, s'utilitza HOG.

### 2.3.1 Caracterstiques de HoG

El procés d'extracció d'una regió consta dels següents passos (veure figura 2.9):

**Pas 1:** El primer pas consisteix en una normalització de la imatge per reduir la influència dels efectes de la il·luminació. A la pràctica, s'utilitza compressió gamma, ja sigui



**Figura 2.9:** Esquema del funcionament de HOG.

calculant l'arrel quadrada o el registre de cada canal de color. La textura de la imatge normalment és proporcional a la superfície local d'il·luminació, per tant, aquesta compressió ajuda a reduir els efectes locals de les ombres i variacions de d'il·luminació.

**Pas 2:** En aquesta segona etapa, es calculen els gradients de primer ordre. Aquestes captures de contorn, silueta, i alguna informació sobre la textura es mantindran tot i les variacions de la il·luminació. S'utilitza el canal de color local dominant, ja que estableix una invariància de color.

**Pas 3:** El tercer pas consisteix en crear una codificació que sigui sensible al contingut de la imatge local sense deixar de ser resistent a petits canvis en la postura o l'aparença.

La finestra de la imatge es divideix en petites regions anomenades 'cel·les'. Per cada cel·la s'acumula un histograma 1-D del gradient o de la orientació dels contorns sobre tots els píxels de la cel·la. Això és el que forma la representació bàsica de l'histograma d'orientació. Cada histograma d'orientació divideix el rang de l'angle del gradient en un nombre fixe de contenidors predeterminats. Les magnituds del gradient dels píxels d'una cel·la són usats per votar en l'histograma d'orientació.

**Pas 4:** Aquest és el pas de la normalització. Analitza grups locals de cel·les que contrasta i normalitza abans de passar al següent pas. La normalització fa que els efectes de la il·luminació, les ombres i els contrastos influeixin menys. Això s'aconsegueix acumulant la mesura d'energia de l'histograma local sobre grups locals de cel·les, que anomenarem blocs. El resultat s'utilitza per normalitzar cada cel·la en el bloc. Normalment, cada cel·la és compartida entre diferents blocs, per cada normalització només depèn del bloc, i per tant, són diferents. Aquesta cel·la pot aparèixer moltes vegades en el vector de sortida final, però cada vegada amb una normalització diferent. Ens referirem als descriptors normalitzats dels blocs com a descriptors HOG.

**Pas 5:** El pas final consisteix en recollir tots els descriptors HOG de tots els blocs de totes les zones sol·lapades de la graella que cobreixen la finestra de detecció en un únic vector numèric, que serà el descriptor de la regió.

## 2.4 Inpainting

Moltes vegades desitjaríem eliminar alguna cosa d'alguna de les fotos que fem, com per exemple un contenidor, una farola, alguna persona, etc., altres vegades senzillament el que es vol és reparar una antiga fotografia, on les cantonades han quedat descolorides. La tasca de completar una imatge o reemplaçar una regió d'una imatge amb informació obtinguda d'altres imatges fent que aquesta modificació no es detecti és el que s'anomena Inpainting.

Hi ha dues estratègies diferents per completar les imatges. La primera consisteix en reconstruir, el més acurat possible, la informació que hi hauria d'haver a la regió que falta, i normalment, els mètodes que utilitzen aquesta tècnica no només utilitzen informació obtinguda de la imatge d'entrada, sinó que obtenen més informació a través d'altres fonts, com poden ser vídeos o múltiples imatges semblants a l'original.

L'alternativa és intentar trobar una manera plausible que permeti omplir els píxels perduts, fent una aproximació de les dades que podrien haver estat allà.

En aquest projecte implementen el mètode proposat per James Hays i Alexei A. Efros [10]. Hays i Efros proposen utilitzar una biblioteca d'imatges per poder completar la imatge original, en canvi, el que proposem nosaltres és la utilització de les imatges de

la Geomòbil, ja que al ser seqüencials, les imatges properes a la imatge d'entrada seran molt semblants i per tant es podrà fer una reconstrucció més acurada.

Hi ha dues raons per expandir la búsqueda de la regió que millor s'adapta a la part eliminada de la imatge original:

1. En moltes ocasions, una regió pot ser impossible d'omplir utilitzant només informació de la imatge d'origen, per exemple, si falta la teulada d'una casa o s'ha eliminat tot el cel, costarà molt d'omplir aquella regió, ja que no hi haurà cap zona semblant a la que falta.
2. Fins i tot, si hi ha contingut de la imatge original que pugui ser utilitzat per omplir la regió que falta, ens podem trobar que hi ha zones que es vegi molt que estan duplicades, com podria ser l'eliminació d'un edifici, i a la reconstrucció, utilitzar un altre edifici de la mateixa imatge, lògicament, en aquest cas, es notaria molt la duplicitat.

No obstant, hi ha uns quants inconvenients a l'hora d'utilitzar aquest mètode. El primer és computacional. Encara que només s'analitzi la imatge d'entrada per trobar les regions que poden omplir la part eliminada, el fet de que es busquin les textures per trobar la millor regió possible, fa que es tardi molt a analitzar una imatge i completar una imatge.

El segon inconvenient és que com més creixi l'espai de cerca, més possibilitats de trobar una regió que s'adapti però que sigui semànticament incorrecte, és a dir, que el contingut de la regió que s'ha trobat és molt diferent a la resta de la imatge original.

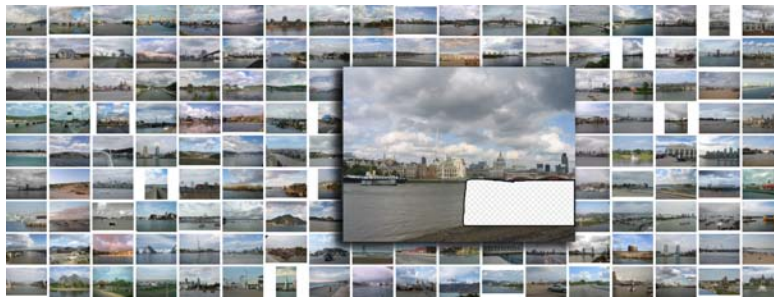
El tercer inconvenient és que les regions que poden servir per omplir la imatge original i que provenen d'altres imatges, és més difícil que tinguin el color exacte i la mateixa il·luminació que l'original. Per tant, es necessiten uns bons mètodes que facin que la imatge resultant no molesti a la vista.

### 2.4.1 Imatges Semànticament Iguals

En el cas del mètode de Hays i Efros, s'utilitza una biblioteca d'imatges, és a dir, tenen una base de dades amb moltes imatges de molts tipus, i mitjançant un descriptor de característiques, seleccionen les que semànticament són més similars a la imatge original, és a dir, que si la imatge original és un paisatge d'una ciutat, a través dels descriptors obtinguts d'analitzar aquesta imatge, seleccionaran les de la base de dades que tinguin uns descriptors semblants, i que per tant, també representaran paisatges de ciutats. A la figura 2.10 es pot veure un exemple.

En el nostre cas, en canvi, en lloc d'utilitzar una biblioteca d'imatges, com que tenim tota la base de dades d'imatges obtingudes per la Geomòbil i sabem que són seqüencials, també sabem que les imatges que més s'assemblaran a l'original seran les capturades





**Figura 2.10:** Exemple de les imatges obtingudes a l'analitzar la base de dades que fa de biblioteca per obtenir imatges semblants a l'original.

immediatament abans i després. A la figura 2.11 es pot veure un exemple. Aquest fet fa que no s'hagin d'analitzar tantes imatges, ja que nosaltres només analitzarem les dues anteriors i les dues posteriors i ella mateixa, per tant, un total de cinc imatges.



**Figura 2.11:** Exemple de les imatges obtingudes per la Geomòbil. La imatge central representa la imatge d'entrada i les altres quatre, les que s'utilitzarien per portar a terme la reconstrucció.

Un cop es tenen les imatges que serviran per obtenir la regió que millor s'adapta a la imatge original, cal trobar aquesta regió.

Primerament cal calcular el context local. Aquest context està format pels píxels que envolten la regió eliminada de la imatge original. Un cop es té el context local, es procedeix a comparar-lo amb les imatges seleccionades anteriorment utilitzant la suma de diferències al quadrat de cada píxel en l'espai de colors  $L * a * b$ .

Només aquelles regions analitzades de les imatges que contenen la totalitat del context local són considerades. Un cop triades, es passa un detector de textures per comprovar que són compatibles i poden omplir la regió eliminada de la imatge original. La regió que

millor s'adapta a la imatge original és la que és copiada a la part eliminada, i posteriorment es difuminen els contorns de la regió per aconseguir que aquest canvi no es noti a simple vista.

## Capítol 3

# Arquitectura de l'Aplicació

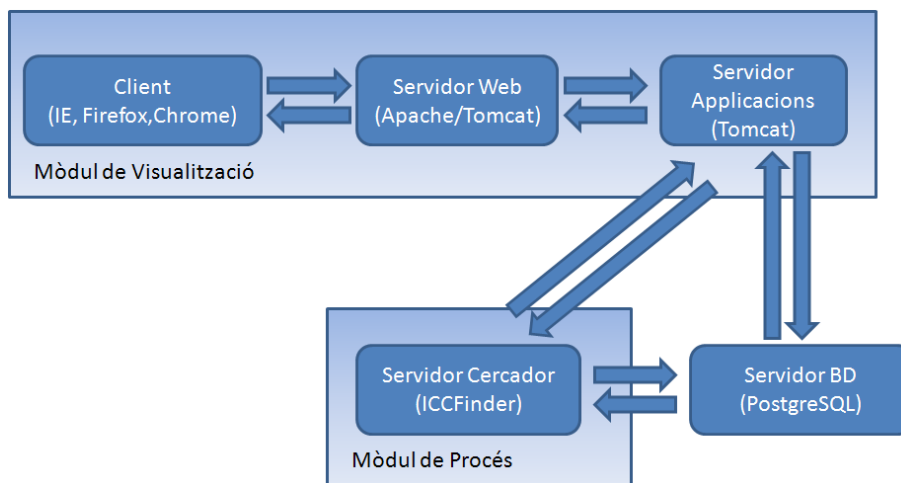
Ja que no s'ha definit l'arquitectura final sobre la qual s'ha d'instal·lar l'aplicació, s'ha optat per fer un disseny el màxim de modular i flexible, per tal que pugui ser adaptat al màxim a diferents escenaris. Amb aquesta finalitat, s'ha dividit l'aplicació en dos grans blocs:

**Mòdul de procés:** En aquest mòdul es concentren tots els mètodes de processament d'imatges i reconeixement. S'ha intentat que aquelles parts més dependents del sistema quedessin el màxim d'aïllades per tal de poder ser substituïdes si hi ha un canvi en l'arquitectura destí de l'aplicació (inicialment s'ha desenvolupat per a Windows).

**Mòdul de visualització:** En aquest mòdul es concentra tota la interacció entre l'usuari i el mòdul de procés. Atès que la part d'interfície acostuma a ser la més dependent de plataforma, s'ha optat per definir-la web, de tal manera que es pot visualitzar el funcionament del mòdul de procés i introduir noves tasques.

Aquests dos blocs es recolzen en un sistema de base de dades, en el qual s'emmagatzema tota la informació. La comunicació entre tots els blocs es fa mitjançant el protocol TCP/IP. A la figura 3.1 es pot veure una visió esquemàtica de l'arquitectura de l'aplicació.

A continuació s'expliquen els dos mòduls en detall, introduint els conceptes necessaris per tal d'entendre els paràmetres dels quals depenen i ajudar a una bona configuració de l'aplicació.



**Figura 3.1:** Esquema de blocs de l'aplicació.

### 3.1 Mòdul de procés

El mòdul de procés és on es processen les imatges per tal d'extreure'n la informació d'interès. S'ha dissenyat aquest mòdul sobre el concepte de *Worker* (treballador), el qual consisteix en un objecte que s'executa en un thread propi. Això permet un grau de paral·lelització molt elevat, però fa necessari establir mecanismes de sincronització i intercanvi d'informació entre aquells *Workers* que estan treballant en tasques relacionades. A la figura 3.2 es mostra el diagrama de funcionament del mòdul de procés. Al principi, es llegeixen els paràmetres de l'aplicació guardats al fitxer *parameters.dat*, a la taula 3.1 es mostren els diferents paràmetres genèrics de l'aplicació. Un cop configurat el servidor, es creen els *Workers* definits en els paràmetres i s'entra en un estat d'hivernació a l'espera d'algun event, durant el qual no es consumeixen recursos del sistema. Si aquest event arriba es processa, i en el cas que sigui un event d'aturada del servidor, es demana a tots els *Workers* que s'aturin. El servidor es queda a l'espera de que tots els *workers* hagin acabat, i finalitza l'execució.

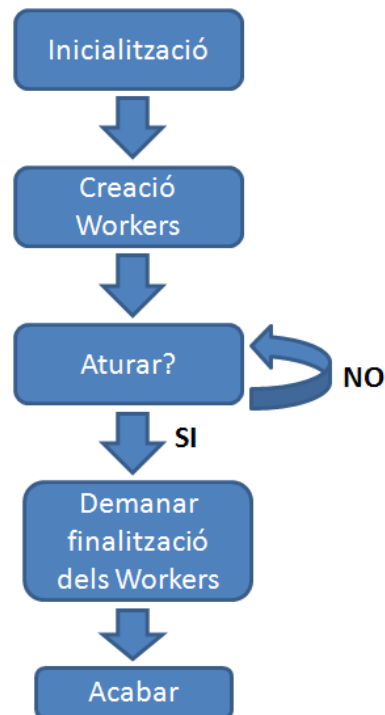
#### 3.1.1 Tasques

Les tasques corresponen a la feina que se li encarrega al servidor. Cada tasca contindrà tota la informació necessària per tal que el *Worker* corresponent pugui realitzar-la. Per exemple, una tasca podria ser trobar les cares en una determinada imatge. El servidor mantindrà en tot moment una llista amb totes les tasques pendents del sistema, i les anirà assignant als diferents *Workers* a mesura que aquests les hi demanin. Moltes vegades, la realització d'un procés requerirà l'execució de més d'una tasca. Aquestes seqüències de

Nom Paràmetre	Descripció
RootPath	Domini en el qual és visible el servidor (URL).
ImagesPath	Clau d'autorització de Google Maps. Cal demanar una clau per al domini en el qual s'instal·li el visualitzador. Quan es demana la clau, la URL indicada ha de coincidir amb el paràmetre GMaps-Domain.
LogFile	Directori en el qual es troben les imatges. Totes les imatges del visualitzador s'accedeixen amb rutes relatives des d'aquest directori.
LogLevel	Indica el nivell mínim que ha de tenir un missatge per a ser guardat en el fitxer de log. Tots els missatges amb valor més elevat a l'indicat es mostraran. Els valors possibles són: <ul style="list-style-type: none"> <li>0 Es mostren tots els missatges. Tots els workers van generant traces del seu funcionament. Aquest nivell permet seguir totes aquestes traces, però si el nombre de workers és elevat, la quantitat d'informació generada pot ser molt gran.</li> <li>1 Es mostren missatges d'accions puntuals. Quan es produeix algun esdeveniment, els workers el comuniquen. En aquest nivell, les traces es redueixen considerablement.</li> <li>2 En aquest nivell només es mostraran els avisos i els errors que es produeixin durant l'execució.</li> <li>3 En aquest nivell només es mostraran els errors.</li> </ul>
WorkersFile	Fitxer que conté la descripció dels Workers. Aquest fitxer és el que confereix al servidor la seva funcionalitat. El format d'aquest fitxer s'especifica més endavant.
DBPars	Paràmetres de connexió amb la base de dades. Aquests paràmetres són els que s'utilitzen tant per al client com per al servidor.
SURFPars	Paràmetres per defecte per a l'extractor de punts característics i descriptors SURF.

**Taula 3.1:** Paràmetres de les aplicacions.

tasques es definiran mitjançant els *Plans*. Els plans indicaran les diferents tasques que es duran a terme per realitzar un procés. Per exemple, si nosaltres volem trobar un determinat tipus d'objecte i guardar-lo a la base de dades, podríem definir un pla consistent en dos

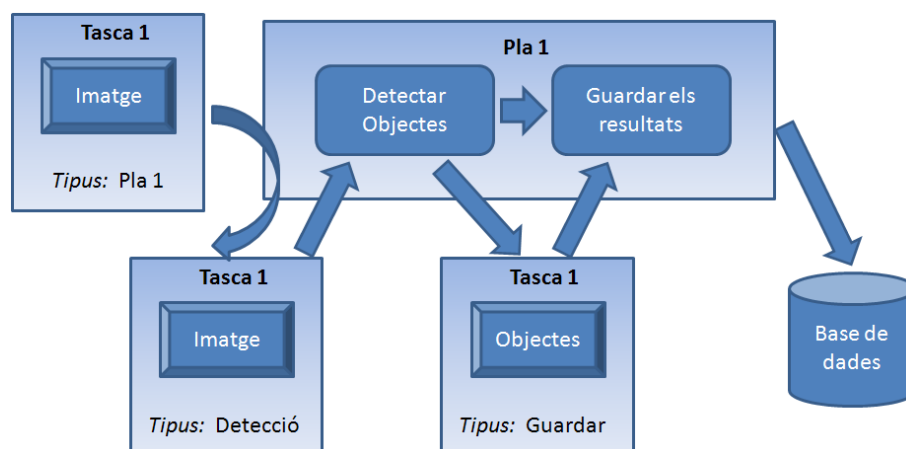


**Figura 3.2:** Diagrama de la vida de l'aplicació.

passos:

1. Detectar els objectes d'una imatge d'entrada. A partir d'una imatge, aconseguir les coordenades dels objectes que s'hi troben.
2. Guardar els objectes detectats a la base de dades. Guardar a la base de dades les coordenades dels objectes que s'ha trobat en el pas 1.

Imaginem que a aquest pla li assignem l'identificador **Pla1**. Ara imaginem que volem detectar i guardar els objectes d'una certa imatge. El que farem serà crear una nova tasca (**Tasca1**), la qual contindrà la imatge en qüestió, i li assignarem com a tipus el pla que volem que l'executi, en aquest cas el **Pla1**. El sistema el que farà és anar convertint aquesta tasca en les noves tasques que requereix el pla. Com el primer pas és la detecció, assignarem a la tasca el tipus detecció i la passarem a un mòdul de detecció. Un cop finalitzat el procés de detecció, el sistema substituirà les dades de la tasca pel resultat i li assignarà el nou tipus, que serà guardar. Això farà que aquesta tasca passi al mòdul de guardar i els objectes detectats es guardaran a la base de dades. Una representació gràfica d'aquest procés es pot veure a la figura 3.3.



**Figura 3.3:** Exemple de la execució d'un pla.

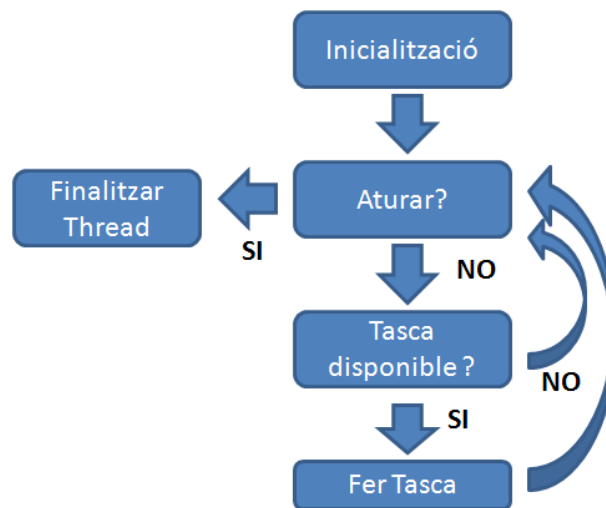
A l'igual que un pla és una seqüència de tasques, també podem crear seqüències de plans per tal de resoldre processos més complexes.

### 3.1.2 Treballadors (Workers)

Els Workers corresponen a un thread de l'aplicació, encarregat de resoldre un tipus concret de tasca. Tots els Workers són especialitzacions d'un tipus genèric de Worker, el qual defineix la base comuna de funcionament i unes propietats generals. El cicle de vida d'un Worker es mostra a la figura 3.4. La primera fase consisteix en la inicialització de totes les propietats i estructures del Worker. Un cop inicialitzat, el Worker entra en un cicle en el qual mentre no se li demani que acabi, va buscant si hi ha alguna tasca disponible que ell pugui fer. En cas que hi hagi una tasca lliure, el Worker la demana i si se li assigna, executa tots els processos necessaris per tal de resoldre aquesta tasca. En cas que no hi hagi cap tasca, es torna a mirar les condicions de parada. Per tal d'evitar un consum excessiu de recursos del sistema, quan un Worker no troba cap tasca disponible, es queda hibernant durant un període de temps que depèn del tipus de Worker i del nombre de cicles sense trobar cap tasca. En qualsevol cas, durant aquesta hibernació, el Worker no consumeix recursos i per tant es poden assignar més recursos a aquells Workers que estan executant tasques.

En la versió actual de l'aplicació, s'han definit els següents tipus de Worker:

**Servidor de Comunicacions:** Aquest Worker s'encarrega de les comunicacions entre el servidor i altres aplicacions, com ara el mòdul de visualització. En començar, el Worker obre un port d'escolta pel qual rep les peticions d'altres aplicacions. Un cop processades les peticions, fa les accions necessàries sobre el servidor (ex. Crear



**Figura 3.4:** Vida d'un worker.

noves tasques) i envia les respostes de les peticions (ex. Llistat de tasques o Workers).

**Controlador de Pla:** Aquest Worker s'encarrega d'anar generant les tasques pertinents per a què tots els passos d'un pla determinat es duguin a terme. Quan es crea un Worker d'aquest tipus, el primer que fa és crear tots els Workers que necessitarà per a resoldre el pla dins el qual es troba. Quan es crea una nova tasca per al pla al qual pertany aquest Worker, el controlador se l'auto-assigna, i genera la tasca o tasques oportunes per resoldre el primer pas del pla. Les tasques creades seran assignades als diferents workers especialitzats dins del pla i cada cop que una tasca del pla finalitzi, el Worker controlador del pla s'encarregarà de crear les noves tasques fins a arribar a l'últim pas del pla i la tasca s'elimina. Quan totes les tasques creades pel controlador per resoldre la tasca inicial s'han acabat, el controlador dona per finalitzada la tasca. Si s'ha especificat un pla posterior, aquesta tasca s'assignarà al controlador del nou pla, sinó serà eliminada.

**Detector d'objectes:** Aquest worker rep una imatge d'entrada, en la qual busca instàncies d'un determinat tipus d'objecte. Els models que ha de buscar s'assignen al Worker en el moment d'inicialitzar-lo. El resultat d'aquest procés és una llista de regions de la imatge.

**Extractor de Descriptors:** Aquest worker rep una imatge d'entrada i la representa mitjançant els seus punts característics. Per a cada punt s'extreu la seva posició i el seu descriptor. El resultat d'aquest procés és una llista de punts i vectors de descriptors.



**IPCA:** Aquest Worker és una implementació de l'Anàlisi Incremental de Components Principals (Incremental Principal Components Analysis). Per tal de fer un bon anàlisi, és necessari tenir una bona estimació de la mitja de les dades. Per aquest motiu, aquest Worker permet dos modes de funcionament: El primer fa la estimació de la mitja i el segon duu a terme l'anàlisi de components principals, utilitzant la mitja per normalitzar les dades. Com a entrada aquest Worker rep un conjunt de vectors.

**Guardador de dades:** Aquest worker s'encarrega de guardar les dades a la base de dades.

## 3.2 Administració del sistema

La part d'administració ens permetrà monitoritzar el funcionament del servidor així com per crear noves tasques. En la figura 3.5 es mostra l'aspecte de la pàgina de control del servidor (ServerStatus.jsp). En aquesta pàgina es mostren tots els Workers i Tasques que estan actives al servidor. Per a tots ells es mostra el seu identificador, quin ha estat el responsable de la seva creació, el seu tipus, el pla al que pertanyen i el seu estat.

Pel que fa a l'estat dels Workers, ens podem trobar amb dues situacions diferents:

**Worker lliure:** Aquest estat es mostra amb un semàfor de color verd. Ens indica que aquest procés actualment no té cap tasca assignada.

**Worker ocupat:** Aquest estat es mostra amb un semàfor de color vermell. Això vol dir que aquest Worker té una tasca assignada i l'està processant. Al costat del semàfor es mostra l'identificador de la tasca que està assignada a aquest Worker.

En el cas de les Tasques ens podem trobar amb tres estats diferents:

**Tasca pendent:** Aquest estat es mostra amb un semàfor de color vermell. Ens indica que aquesta tasca encara no ha estat assignada a cap Worker.

**Tasca en procés:** Aquest estat es mostra amb un semàfor de color amber. Això vol dir que aquesta tasca ha estat assignada a un Worker, però que encara no ha finalitzat.

**Tasca finalitzada:** Aquest estat es mostra amb un semàfor en verd, que indica que la tasca ha finalitzat i per tant els seus resultats estan disponibles.

Finalment, a la part inferior de la pàgina es mostra un petit formulari en el qual es permet afegir un directori i l'identificador d'un pla. Aquest formulari permet afegir una nova tasca al sistema. Es processaran tots els fitxers dins de l'arbre de directoris que penja del directori que es passa com a paràmetre. Cal tenir en compte que el nom de directori s'agafa relatiu al directori d'imatges del servidor.

localhost ( 127.0.0.1 )

Workers				
ID	Status	Parent	Plan	Type
0		-1	-1	WORKER_TASK_COMMUNICATION_SERVER
1		-1	10	WORKER_TASK_PLAN_CTRL
2		1	11	WORKER_TASK_PLAN_CTRL
3		-1	12	WORKER_TASK_PLAN_CTRL
4		-1	13	WORKER_TASK_PLAN_CTRL
5		4	13	WORKER_TASK_DESC_EXTRACTOR
6		1	10	WORKER_TASK_RECOGNITION
7		3	12	WORKER_TASK_DESC_EXTRACTOR
8		21	2	WORKER_TASK_DESC_EXTRACTOR
9		4	13	WORKER_TASK_DESC_EXTRACTOR
10		1	10	WORKER_TASK_RECOGNITION
11		2	11	WORKER_TASK_DESC_EXTRACTOR
12		3	12	WORKER_TASK_DESC_EXTRACTOR
13		1	10	WORKER_TASK_RECOGNITION
14		3	12	WORKER_TASK_IPCA
15		4	13	WORKER_TASK_DATA_STORING
16		4	2	WORKER_TASK_DESC_EXTRACTOR
17		3	12	WORKER_TASK_DATA_STORING
18		1	10	WORKER_TASK_DATA_STORING
19		3	2	WORKER_TASK_DESC_EXTRACTOR
20		6	2	WORKER_TASK_DESC_EXTRACTOR
21		16	2	WORKER_TASK_DESC_EXTRACTOR
22				WORKER_TASK_DESC_EXTRACTOR
23				WORKER_TASK_DESC_EXTRACTOR
24				WORKER_TASK_DESC_EXTRACTOR
25				WORKER_TASK_DESC_EXTRACTOR
26				WORKER_TASK_DESC_EXTRACTOR
27				WORKER_TASK_DESC_EXTRACTOR
28		18	2	WORKER_TASK_DESC_EXTRACTOR
29		15	2	WORKER_TASK_DESC_EXTRACTOR
30		12	2	WORKER_TASK_DESC_EXTRACTOR
31		11	2	WORKER_TASK_DESC_EXTRACTOR
32		11	2	WORKER_TASK_DESC_EXTRACTOR
33		14	2	WORKER_TASK_DESC_EXTRACTOR
34		13	2	WORKER_TASK_DESC_EXTRACTOR
35		20	2	WORKER_TASK_DESC_EXTRACTOR
36		2	11	WORKER_TASK_IPCA
37		2	11	WORKER_TASK_DATA_STORING

Tasks				
ID	Parent	Plan	Status	Type
1	-1	11		WORKER_TASK_PLAN_CTRL
2	1	11		WORKER_TASK_DESC_EXTRACTOR
3	1	11		WORKER_TASK_DESC_EXTRACTOR
4	1	11		WORKER_TASK_DESC_EXTRACTOR
5	1	11		WORKER_TASK_DESC_EXTRACTOR
6	1	11		WORKER_TASK_DESC_EXTRACTOR
7	1	11		WORKER_TASK_DESC_EXTRACTOR
8	1	11		WORKER_TASK_DESC_EXTRACTOR
9	1	11		WORKER_TASK_DESC_EXTRACTOR
10	1	11		WORKER_TASK_DESC_EXTRACTOR
11	1	11		WORKER_TASK_DESC_EXTRACTOR
12	1	11		WORKER_TASK_DESC_EXTRACTOR
13	1	11		WORKER_TASK_DESC_EXTRACTOR
14	1	11		WORKER_TASK_DESC_EXTRACTOR
15	1	11		WORKER_TASK_DESC_EXTRACTOR
16	1	11		WORKER_TASK_DESC_EXTRACTOR
17	1	11		WORKER_TASK_DESC_EXTRACTOR
18	1	11		WORKER_TASK_DESC_EXTRACTOR
19	1	11		WORKER_TASK_DESC_EXTRACTOR
20	1	11		WORKER_TASK_DESC_EXTRACTOR
21	1	11		WORKER_TASK_DESC_EXTRACTOR

Task Manager	
File Path:	<input type="text"/>
Plan ID:	<input type="text"/>

Figura 3.5: Informació mostrada pel mòdul de seguiment de tasques i Workers.

### 3.3 Base de dades

L'últim element de l'aplicació és la Base de Dades. Tot i que l'aplicació s'ha desenvolupat pensant en un servidor basat en PostgreSQL, la part d'accés a dades s'ha aïllat en una sola classe i tots els accessos es fan mitjançant comandes SQL, el que permet una fàcil migració cap a altres sistemes de base de dades per part del servidor. En el cas del visualitzador s'utilitza JDBC, disponible per la gran majoria dels sistemes de bases de dades. Per tant, canviant el driver d'accés a la base de dades hauria de permetre connectar-se amb qualsevol sistema gestor de base de dades.

A la base de dades es guarda la informació resultant dels processos executats en el servidor i tota la informació necessària per a la correcta visualització d'aquestes dades. En un futur es pot considerar incloure també tots els paràmetres que ho permetin. A nivell de continguts, la base de dades consta de les següents taules:

**Imatges:** En aquesta taula es guarda la informació de totes les imatges capturades per la Geomobil, com ara la posició de la Geomòbil en el moment de capturar aquesta imatge, o el directori on es troba guardada aquesta imatge (si està disponible). Aquestes dades provenen dels fitxers d'orientacions de la Geomòbil.

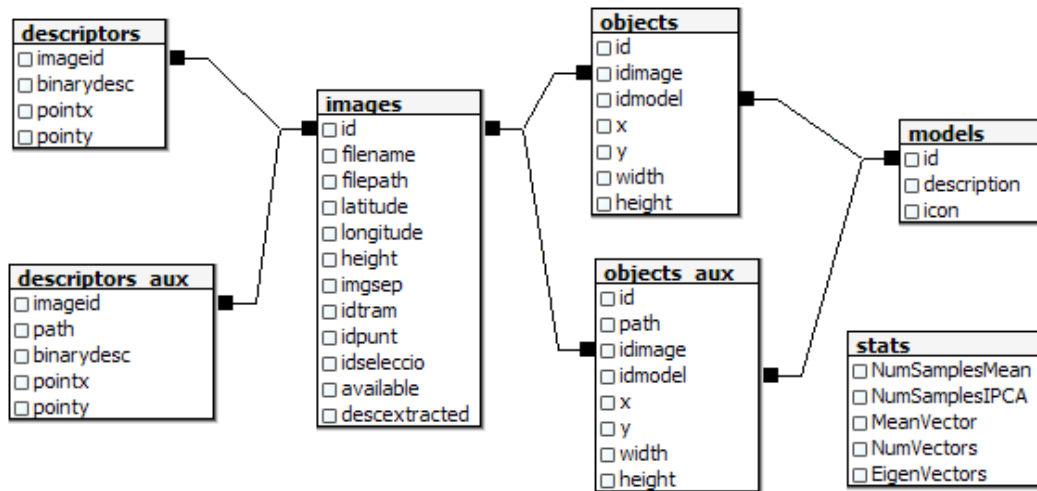
**Models:** Taula on es mostra la informació sobre els diferents models que es consideren. Ens permet definir quins tipus d'objectes ens interessarà buscar.

**Objects:** En aquesta taula es guarden els objectes que s'han trobat. Cada objecte s'associa a la imatge on s'ha trobat i al model al que pertany aquest objecte.

**Descriptors:** Taula on s'emmagatzemen els punts d'interès i els corresponents descriptors binaris que representen cada imatge.

**Stats:** En aquesta taula es guarda la mitja de les dades i els vectors propis extrets a partir dels descriptors de les dades i que ens permeten crear els codis binaris.

A més a més de les taules principals, donat que es va veure que en alguns casos les dades d'ubicació de les imatges en els fitxers d'orientacions, no corresponien exactament amb la ubicació real de les imatges, s'han afegit dues taules auxiliars per als resultats, que permeten emmagatzemar els resultats en aquells casos en què no s'ha trobat la imatge a la taula Imatges, i per tant no es pot obtenir l'identificador que permet associar els objectes i descriptors amb la seva imatge. Aquestes taules es diuen Objects\_aux i Descriptors\_aux, les quals són idèntiques a les taules Objects i Descriptors, amb la excepció que s'ha incorporat un camp "path", on es guarda la ruta a la imatge, i ens permetrà recuperar els resultats un cop identificada la imatge a la qual corresponen. L'esquema de la base de dades i la relació entre els diferents atributs es pot veure a la figura 3.6.



**Figura 3.6:** Esquema de la base de dades.

# Capítol 4

## Resultats

Un cop definida la metodologia, en aquest apartat es presenten els resultats obtinguts tant en el cas de la detecció de cares i persones, com per a les matrícules. El mètode de l'Inpainting, degut al temps que tarda a processar totes les imatges, no s'ha pogut fer un càlcul dels resultats en totes les imatges on s'han detectats persones.

Els resultats s'han extret mitjançant una verificació manual dels resultats obtinguts d'una seqüència de la Geomòbil que conté 982 imatges. En aquestes imatges hi trobem un total de 479 cares i 588 vianants. Aquesta diferència entre el nombre de cares i el de vianants és perquè, tal i com s'ha comentat anteriorment, encara que algunes cares no es consideren reconeixibles, si tota la persona és visible encara pot ser identificable. En algunes imatges és possible trobar més d'una cara o persona, per tant, ens podem trobar que haguem de tractar més d'una regió per imatge.

La detecció de matrícules s'ha fet sobre un conjunt seleccionat d'imatges, donat que la proporció d'imatges en que apareixen matrícules és molt baixa. S'ha seleccionat un total de 30 imatges on apareixen 45 vehicles amb la matrícula visible.

Per al càlcul de les estadístiques, s'han definit dos barems diferents:

**1r borem:** És el més restrictiu. Totes les cares que estan borroses, que estiguin en un angle molt tancat (és a dir, quan estan d'esquenes a la càmera però amb la cara girada), que portin gorra o barrets i ulleres de sol o que estiguin cremades i que no s'han detectat, es consideren errors.

**2n borem:** És el més permissiu. Les cares descrites anteriorment, quan es creu que no són identificables, no es detecten com a error.

En la detecció de persones també s'han utilitzat aquests dos barems.

## 4.1 Detecció de cares

Els resultats obtinguts en aquesta secció estan representats a la taula 4.1. Si analitzem els resultats utilitzant el barem més restrictiu, torbem un 51.35% d'encerts, és a dir, que detecta 246 de 479 cares, mentre que si utilitzem el segon barem, trobem un 67.77% d'encerts, és a dir, que detecta 362 cares. Tenint en compte que l'objectiu és evitar el reconeixement de les persones que apareixen en les imatges, el 2n barem no sembla un mal indicador, ja que en el primer algunes de les imatges que es perden estan prou degradades com per a dificultar-ne el reconeixement de la persona que hi apareix.

**Taula 4.1:** Resultats de la detecció de cares

	<b>1r barem</b>	<b>2n barem</b>
Encerts	51.35%	67.77%
Número de cares	246/479	362/479
Falsos positius	6.09/cara	4.14/cara

A la figura 4.1d, podem veure com la gorra tapa parcialment la cara de la persona. En aquest cas, si utilitzéssim el primer barem per fer la classificació, diríem que es un fals negatiu, ja que no s'ha trobat la cara, en canvi, amb el segon barem, l'acceptariem com a negatiu, ja que al no veure's tota la cara, es considera que no es pot fer la seva identificació.

A la figura 4.1a i a la figura 4.1e hi ha un exemple del que passa amb el perfil, on a la segona imatge no es detecta cap cara, ja que el grau de gir de la cara és molt poc, i en aquest cas, es considera com un fals negatiu si es classifica amb el primer barem i correcta si es classifica amb el segon. En canvi, a la primera, com que el grau de gir és més gran i es pot veure una mica més la cara, aquesta ja es troba.

Un altre cas molt comú en les imatges és el de les imatges cremades. En aquests casos, normalment no es detecta i la majoria de vegades es compta com a ben classificada, ja que el mateix cremat de la cara ha difuminat les faccions de la persona fent que pràcticament sigui irreconeixible. Un exemple és el que trobem a la figura 4.1c.

Al buscar cares amb mides molt petites, s'està incrementant considerablement el nombre de falsos positius. En el cas de les cares, el nombre total de falsos positius trobats és de 1499. El que significa una mitjana de 1.52 falsos positius per imatge, o de 6.09 falsos positius per cada cara detectada amb el primer barem o 4.14 pel segon. Tot i que el nombre de falsos positius és gran, aquests valors han quedat afectats per algunes imatges com la de la figura 4.1f, on apareix una reixa amb uns adornaments que semblen cares, i el detector els detecta múltiples vegades.



**Figura 4.1:** Exemples de detecció de cares. (a) i (b) Deteccions de cares bones. (c) Imatge cremada amb un fals negatiu. (d) Cara amb complements. (e) Cara amb un angle molt tancat. (f) Mala detecció, en aquest cas degut a que la reixa té formes similars a una cara.

## 4.2 Detecció de persones

Utilitzant els mateixos criteris i la mateixa seqüència que en la detecció de cares, per al detector de vianants trobem 55.10% d'encerts per al primer barem, és a dir, que detecta 324 de 588 persones, mentre que si utilitzem el segon barem, trobem un 70.13% d'encerts (450/588). A la taula 4.2 podem veure els resultats obtinguts.

A la figura 4.2 es mostra un exemple de detecció de persones.

Donat que el detector de persones està entrenat amb persones de peu, aquelles persones que apareixen assegudes generalment no es detecten, el que es veu reflectit en la



**Taula 4.2:** Resultats de la detecció de persones

	1r barem	2n barem
Encerts	55.10%	70.13%
Nombre de persones	324/588	450/588
Falsos positius	6.04/persones	4.36/persones

diferència obtinguda entre el primer i segon barem. A la figura 4.2b es mostra un exemple d'una persona asseguda que no es detecta correctament.



(a)



(b)

**Figura 4.2:** Exemples de detecció de persones. (a) Detecció correcta de les persones. (b) Mala detecció de les persones.

En el cas del detector de persones, també tenim un índex de falsos positius elevat. En total hi ha 1964 falsos positius, és a dir, 2 falsos positius per imatge, o 6.04 falsos positius per persona trobada amb el primer barem o 4.36 amb el segon.



## 4.3 Detecció de matrícules

Tal i com s'ha dit anteriorment, la detecció de matrícules s'ha aplicat sobre un conjunt més limitat d'imatges. Un exemple de detecció es pot veure a la figura 4.3.

Analitzant els punts on falla el detector, s'ha comprovat que moltes de les imatges on es poden veure matrícules corresponen a xamfrans, on hi ha una forta deformació de les imatges, i les matrícules apareixen rotades. A la figura 4.3c es pot veure aquest efecte, i com el detector de text no pot trobar cap de les matrícules que hi apareixen. Si agafem la regió que conté les matrícules i en desfem la rotació, el detector en pot detectar la majoria, com es veu a la figura 4.3d. Si es corregeixen les rotacions de les matrícules, el percentatge d'encert puja fins al 64.4%, detectant 29 de les 45 matrícules.

També hi ha algun cas de matrícules antigues, les quals tenen molt pocs dígit, i per no es detecten com a regions de text. Un exemple d'aquest cas es veu a la figura 4.3b, on podem veure que la matrícula normal es detecta correctament, però que la matrícula antiga s'ha perdut.

## 4.4 Inpainting

En aquesta part no hi ha resultats numèrics ja que el temps que tarda a executar és molt elevat. A la figura 4.4 es pot veure el resultat després de la implementació de tot el sistema, es a dir, es mostre el procés que segueix la imatge per aconseguir la seva anonimatització. La figura 4.4a és la imatge original, tal i com la trobem a la base de dades de la Geomòbil. La següent imatge és la resultant de la detecció de persones. La persona queda requadrada en vermell. La figura 4.4b és el resultat d'eliminar la regió obtinguda amb el detector de persones, i finalment, a l'última imatge hi tenim el resultat obtingut després d'aplicar l'Inpainting.

A la figura 4.5 hi ha un exemple d'un resultat dolent utilitzant aquest sistema. En aquest cas hi ha moltes regions on s'han detectat persones, algunes d'elles són falsos positius, i això fa que el resultat obtingut no sigui bo, ja que s'han de completar moltes regions de la imatge i per tant, la probabilitat de que la imatge resultant sigui dolenta, és molt elevada. En aquest cas, a més a més, es pot veure com les regions escollides per completar la imatge no són les millors, ja que en elles hi apareixen persones, per tant, no s'elimina la informació amb contingut sensible.



(a)



(b)

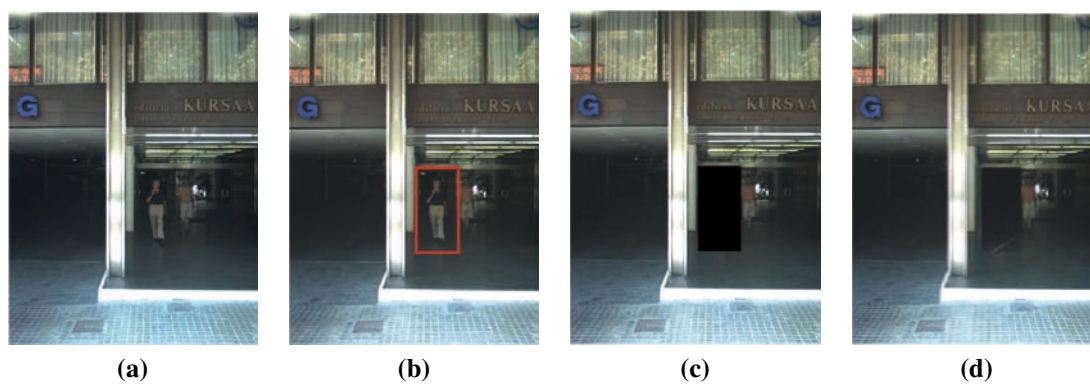


(c)

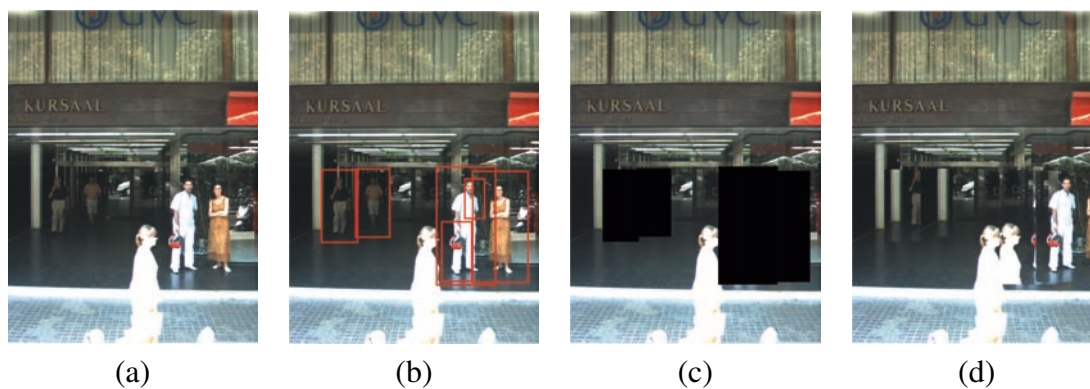


(d)

**Figura 4.3:** Exemples de detecció de matrícules. (a) Correcta detecció d'una matrícula, també es veu com en matrícules antigues, al ser una regió poc ampla, no es detecta. (b) Exemple d'un error en la detecció de matrícules. (c) Exemple d'un error en la detecció de matrícules. (d) Regió corresponent a la figura 4.3c, on s'ha eliminat la rotació. En aquest cas les matrícules sí que es poden detectar.



**Figura 4.4:** Procés del sistema amb aplicació de l'Inpainting



**Figura 4.5:** Un exemple de resultats dolents utilitzant l'Inpainting.



## Capítol 5

# Conclusions i treballs futurs

### 5.1 Conclusions

En aquest projecte s'ha presentat un anàlisi dels problemes causats per la presència de persones i matrícules a imatges a disposició del públic. També s'ha vist que en casos on les persones tenen una resolució baixa, quan el cos es totalment visible amb el context de la imatge es pot reconèixer a la persona.

S'ha presentat un sistema per tractar amb eficiència els problemes de privacitat utilitzant els clàssics detectors de cares i vianants i una variant dels detectors de textos per poder trobar les matrícules. En alguns casos, les regions han estat eliminades en lloc de difuminades, com és el cas de les regions corresponents a la detecció de persones.

Finalment, aquestes regions que s'han eliminat de les imatges, s'han reemplaçat per regions similars d'altres imatges utilitzant el mètode de l'Inpainting. Com que treballem amb imatges seqüencials, s'aprofita la redundància temporal per trobar les escenes que millor s'adapten per a poder portar a terme la reconstrucció de la imatge.

Encara que els resultats obtinguts són prometedors, ens hem trobat amb problemes a l'hora de cercar imatges on trobar les regions òptimes pel procés de l'Inpainting, ja que s'han obtingut reconstruccions dolentes. Aquest problema es pot resoldre mitjançant un conjunt més extens d'imatges, o aprofitant que les imatges estan georeferenciades, utilitzar la redundància espacial per tal de trobar altres imatges capturades prop de l'escena analitzada.

També cal dir que un altre dels problemes trobats ha estat a les matrícules, ja que quan aquestes apareixen rotades el sistema no les detecta.

## 5.2 Treballs futurs

Com a treballs futurs d'aquest projecte es podria aplicar un nou sistema de detecció de matrícules basat en dígit que permetés detectar-les encara que estiguin rotades, que és quan més falsos positius tenim.

Com que s'han utilitzat els mètodes clàssics de detecció de cares i persones, es podrien utilitzar altres mètodes per aconseguir un nombre d'encerts més elevat, per tant, s'haurien de tenir en compte els diferents complements que porten les persones, sobretot pel que fa a la detecció de cares, ja que quan més falsos positius dona és quan la persona porta gorra, ulleres o bigoti. Per a les persones, caldria millorar-lo per detectar quan estan assegudes i en altres posicions, ja que el que hi ha implementat, majoritàriament només detecta les persones dretes.

I finalment, la part que necessita més actualitzacions és la relacionada amb l'Inpainting, ja que es podrien buscar altres tècniques de reconstrucció, com pot ser omplir la regió eliminada amb regions més petites en lloc d'utilitzar-ne una de sola, ja que d'aquesta manera es podria obtenir una imatge final més semblant a l'original. A més a més, per millorar aquest mètode es podrien buscar altres mètodes de detecció de regions i la manera d'eliminar els contorns d'aquesta per adaptar-la correctament a la imatge a reconstruir.

# **Annex**

Article enviat al Congrés Català d'Intel·ligència Artificial que es celebrarà a Tarragona durant el mes d'octubre.

# Preserving Privacy in Public Available Urban Image Database using Inpaintig

Camp DAVESA <sup>a</sup>, Xavier BARÓ <sup>a,b</sup> and Jordi VITRIÀ <sup>a,c</sup>

<sup>a</sup> *Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Barcelona*

<sup>b</sup> *Universitat Oberta de Catalunya, Rambla del Poblenou 156, 08018, Barcelona*

<sup>c</sup> *Dept. Matemàtica Aplicada i Anàlisi, Gran Via 585, 08007, Barcelona*

## Abstract.

In the last years the number of publicly accessible image databases, such as the one available in Google Street View, have been largely increased. The presence of recognizable people, license plates, and other personal information in these images, entails privacy problems in many countries. Current methods aimed to preserve people privacy in those images, are commonly focused only on the detection and blurring of their faces. However, in most of the cases persons remains recognizable if enough context appears in the scene. In this paper, we propose a system which use face and pedestrian detectors in order to remove sensible regions of the image. Finally, an inpainting method is used in order to restore the removed regions, obtaining a user-friendly image without privacy problems.

**Keywords.** Inpainting, HOG, Haar, privacy

## Introduction

In the last few years there has been a rapid increase of the amount of urban images available on the web. Although the largest and probably the most well-known collection is the Google Street View, our system is based on images captured by the Geomobil[1], a mobile mapping system developed by the Institut Cartogràfic de Catalunya<sup>1</sup>. This system consists of a van with a set of six calibrated cameras, which are synchronized with a GPS/INS system. Using the Geomobil, more than 10km of Barcelona streets where captured and georeferenced.

Those images represents a great source of information for city management tasks, navigation purposes or for being used by computer vision research community. Nevertheless, since these images were captured in real conditions, most of them contains people walking, cars, and other sensible information which needs to be considered before publishing the dataset, and the privacy of the people must be guaranteed.

A common approach to protect the privacy of people is to use a face detection software which is aimed to blur faces making them unrecognizable. Face detection is one of the most studied object detection problems in the literature. Variations in facial appearance, lighting, expressions, and other factors transforms the face detection problem in

---

<sup>1</sup>[www.icc.cat](http://www.icc.cat)



a complex pattern recognition problem, where learning-based approach constitutes the most effective strategies for constructing face/non-face classifiers [2]. One of the most extended methods for face detectors is the one proposed by Viola & Jones in [3].

An example of application of the privacy protection is found in the Google Street View [4] mentioned above. They blur faces and license plates found in the images. They have a 89.0% of hits in finding faces, and 96.5% for license plate from US and 93.6% for license plates from EU. The differences between their method and our method is that they use a post-process using neural networks.

The use of face detection methods have two main shortcomings. Firstly, that some faces are too small to be detected for current methods, whereas they can be recognizable for another person. Second, that even when the face has been blurred, a person can be recognized by conspicuous characteristics of the rest of the body. In order to deal with that problem, we propose a system with a face detector and a people detector. The face detector used is based on the method by Viola & Jones and returns the face regions of images that will be blurred. The people detector based on the algorithm of Dalal & Triggs [5] is used to analyze the images to find people regions, but in this case, if people are blurred, it will distort the image, so the regions will be removed and an inpainting method will be used to restore the image with making less impact.

The paper is organized as follows: In Section 1, face detector, people detector, and inpainting methods are described. Section 2 explains how the system works. Section 3 shows the results for face detection, people detection and inpainting. Finally, Section 4 concludes the paper.

## **1. Methods**

In this section the main methods used in order to preserve the privacy of people appearing in images are described. The system architecture and implementation details are described in future sections.

### *1.1. Face detection*

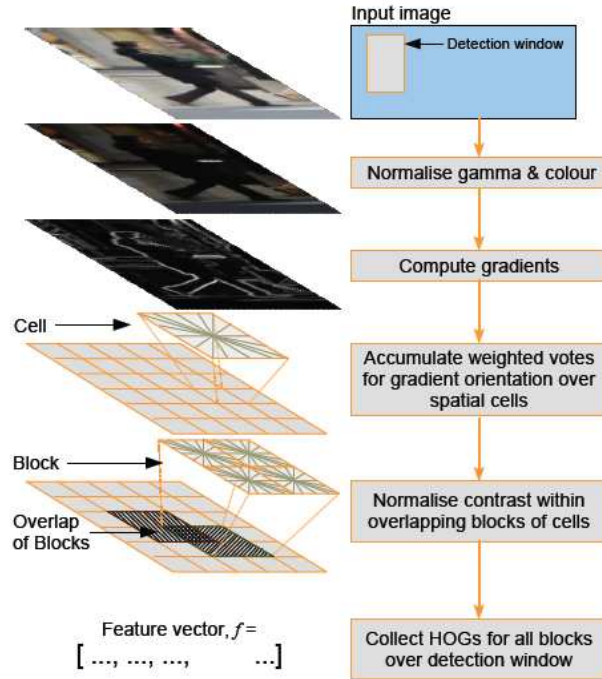
In this work we use the implementation of Lienhart & Maydt [6], who used an extended of the Viola & Jones original Haar-like feature set.

In that approach, AdaBoost [7] algorithm is used in order to learn an ensemble of easily learnable weak classifiers, which are linearly combined into a single strong classifier. Each weak classifier consists of a decision stump, which is a combination of a Haar-like feature and a decision threshold value.

Detection is performed using a sliding window, which analyzes the whole image at different positions and sizes. Each window is then classified as face or non-face. In order to allow real-time, a cascade architecture is used and Haar-like features are optimally calculated using integral images.

### *1.2. Pedestrian detection (HOG)*

People detection in images is a hard task because it is a deformable object with a great variability in his appearance. The detection of people has been studied extensively in security and in driving assistance, and the technique seems to give better results in the



**Figure 1.** Scheme of HOG descriptor extractor. [8].

literature is the use of Histograms of Oriented Gradients (HOG), introduced by Dalal & Triggs [5]. Similarly to the case of face detection, the input images is analyzed using a sliding window, but in this case, instead of being a square, the window keep the proportions of human body. As shown in the figure 1, the process has the following parts:

1. Normalize the image to reduce the illumination effects as shadowing and illumination variances.
2. Calculate the first order gradient to capture the contour, silhouette, and some texture information to provide robustness to illumination variance.
3. The image window is divided into small regions named "cells". For each cell, a 1-D HOG is calculated.
4. Group cells in blocks and normalize the contrast of each cell in the block. The normalized descriptor blocks are referred as Histogram of Oriented Gradient descriptors.
5. Collect the HOG descriptors from all blocks.

Once the region has been described using HOG, Support Vector Machines (SVM) are used in order to perform the final decision.

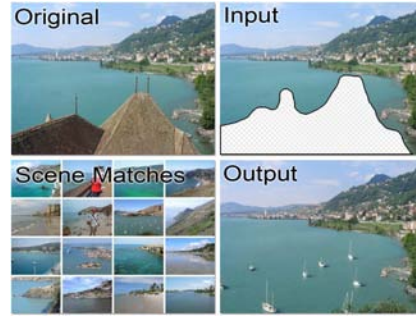
### 1.3. Inpainting

Inpainting is a method used to restore images that have some removed regions. Image completion is the task of filling in or replacing an image region with new image data such that the modification can not be detected.

The method used in this article is based on the work made by Hays and Efros [9]. As shown in Algorithm 1, to reconstruct the input images, a set of semantically equal images to the input one are needed to find the region that fits best in the hole. In our case, as images are captured sequentially, we will use them to find that region, because they are semantically equal.

When images that could contain the best region to restore the input image are chosen, a local context is defined by using the pixels contained in a radius around the boundary of the deleted region. This context is compared against all the selected images using pixel-wise SDD error, and only regions for which the context is fully contained are considered. We also calculate the texture of the regions to ensure a better fitting.

Once the best region is selected, it is used to restore the input image, and boundaries of the region are faded for having a better result. In figure 2 there is an example obtained by Hays and Efros.



**Figure 2.** An example using the method by Hays and Efros [9].

---

**Algorithm 1** Inpainting [9]

---

**Require:**

Input image  $\mathcal{I}$  without some regions and an image Database

Select images from the database that are semantically equal to  $\mathcal{I}$

**for** every region **do**

    Calculate the local context of each deleted region from  $\mathcal{I}$

    Compare the local context against all selected images

    Compute a texture similarity descriptor in the selected region to know if it fits in  $\mathcal{I}$

    Select the region that best fits in  $\mathcal{I}$

    Copy the selected region in the deleted region from  $\mathcal{I}$

    Fade the boundaries of the region

**end for**

---

## 2. System

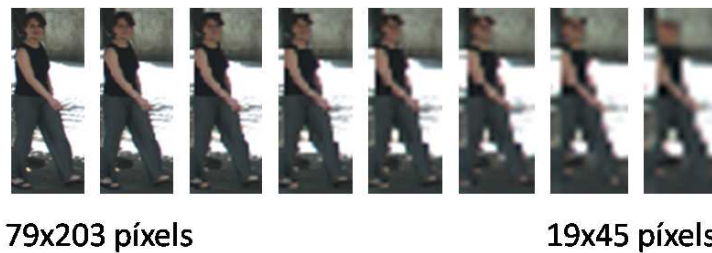
To delete faces and people we need to locate them in the images and decide the size of each that is considered recognizable. Once localized the regions that we want to hide, we must apply methods to delete the contents of these regions.

In most of images from Geomobil appear pedestrians walking the streets. The size of those pedestrians differs greatly depending on the distance between them and the camera, and besides, many of the faces are shown in profile. In figure 3 there are two scenes where pedestrians appear in different distances with the size of their faces. On the left image we can see a small face around  $11 \times 11$  pixels that is not recognizable, and in the right image all faces are recognizable. Based on that we decided that the minimum size of a face for being recognizable is around  $18 \times 18$  pixels.



**Figure 3.** On the left, we can see a face that we do not consider identifiable. On the right there are some identifiable faces. The size in pixels is shown in each case.

As in face, we must set a size where we decide that the pedestrian is no longer identifiable. In figure 4 a pedestrian is showed gradually changing its resolution. Notice that the face is approximately 10% of the height of the person, we consider that if we see the complete person, the face its around  $12 \times 12$  pixels.



**Figure 4.** From left to right, the pedestrian pixel size is decreasing. Around a 120 pixels height the person becomes difficult to identify.

Once the sizes are decided, the process to anonymity the images can start. the system starts using the people detector method using HOG and the face detector method based on Viola & Jones algorithm, both described in Section 1.

Then the system deletes the people regions and blurs the face regions. Finally, the system uses inpainting to restore the images that have a region that was deleted. To use inpainting is necessary to find images semantically similar as the input image.

To do it, it is necessary to define a local context with a 80 pixel radius of the boundary of the pedestrian region. This context is compared against all selected images using pixel-wise SSD error in  $L^*a^*b$  color space. Only regions for which the context is fully contained in the matching scene are considered. The texture of the regions are also calculated to ensure a better fitting.

When all this process is done, the result is an image without identifiable people.

### 3. Results

In order to present the results, first we discuss the data, validation and experiments.

- *Data:* We use a set of 982 images, where there are a total of 479 faces and 588 pedestrians. This difference between the number of faces and pedestrians is because although some faces are considered not recognizable, if the complete person is visible, it can be identified. In many images there are more than one face or person, so it is possible to find more than one region per image to be blurred or deleted.
- *Validation:* The results have been extracted using a manual verification of them. To calculate the estadistics, we defined two different criterion:
  - \* **1st criteria:** This is the most restrictive. All blurred faces, faces in a very tight angle, wearing a cap or a hat or sunglasses or burned faces that have not noticed, are considered errors.
  - \* **2nd criteria:** This is the most permissive. The faces described above, if we consider that it is not possible to be identificates, are not considered as an error.
 This criteria is used in people detection.
- *Experiments:* We have three diferent experiments. The first one is for face detection, the second one, is for people detection and third is for inpainting.

#### 3.1. Face detection results

The results obtained in this section are showed in table 1. In our case, an accuracy near 70% is considerably high given the difficulty of the problem and the results reported in the literature. Examples of detection are showed in figure 5.

As you can see in table 1, there are some false positive face detections, in some cases it is because in the image there are some regions with a similar structure as a face, but they are not. In figure 5f there is an example, and that increases the false positives.

**Table 1.** Face detection results

	1st criteria	2nd criteria
Hit rate	51.35%	67.77%
Faces number	246/479	362/479
False positives	6.09/face	4.14/face

In other cases, images do not have a good quality, some times they are burned and face are not detected. An example of that is shown in figure 5c.



**Figure 5.** Examples of face detection. (a) and (b) Correct face detections. (c) Burned image with a false negative. (d) Face with complements. (e) Face in a close angle. (f) Wrong detections because the gate has similar shapes as a face.

### 3.2. People detection results

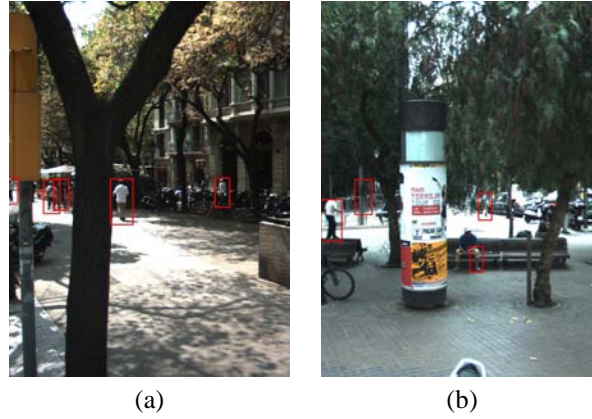
For this experiment we used the same criteria and the same set of images as in face detection. The results obtained in this section are showed in table 2.

**Table 2.** People detection results

	1st criteria	2nd criteria
Hit rate	55.10%	70.13%
People number	324/588	450/588
False positives	6.04/people	4.36/people

In this case we have around 70% of hit rate, as in face detection, and it is considerably high given the difficulty of the problem, because we can find people in many different positions and shapes. Examples of detections are showed in figure 6.

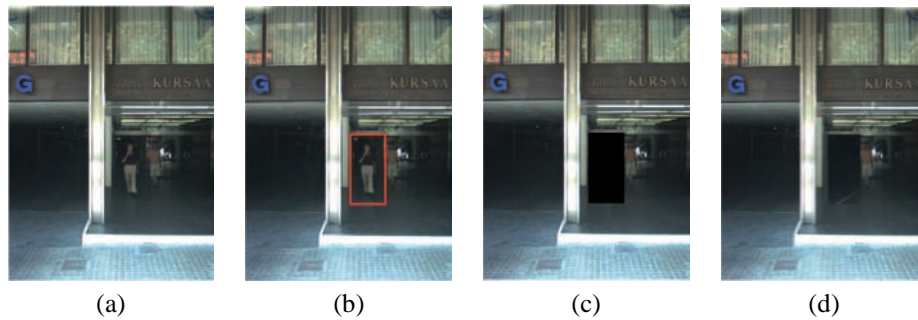




**Figure 6.** Examples of people detection. (a) Good people detection. (b) Bad people detection.

### 3.3. Inpainting results

In this part there are no numeric results, because of the time that it cost to execute. In figure 7 there is a result when all the system is implemented. Figure 7a is the original input image, next there is the same image with people detection. In figure 7c the people regions are deleted from the image, and finally, there is the result.



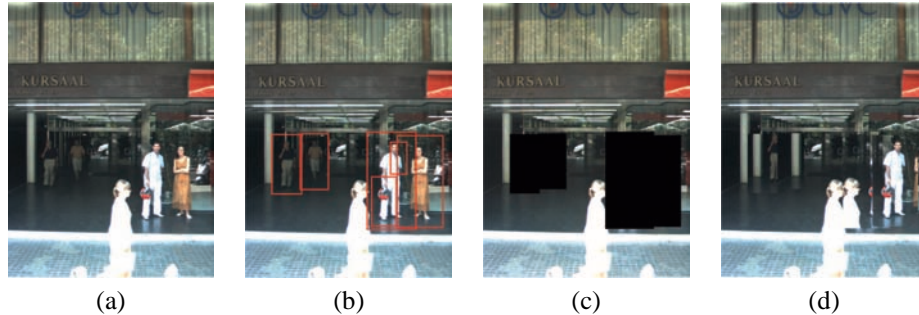
**Figure 7.** Process of inpainting algorithm

In figure 8 there is an example of a bad result using the system. In this case, there are lost of people regions, some of them are false positive, and it produce a bad result. Besides, in this case, regions selected to fill in regions are not the optimums and people appears on them.

## 4. Conclusions and future work

An analysis of the privacy problems due to people appearing in urban image databases has been presented, where we stated that people can be recognized at very low resolution when the whole body is provided and enough context is available.

A system to effectively address the problem of people privacy has been presented, using classical face and pedestrian detectors, sensible regions of the images are removed instead of blurring only faces as is done in most known systems.



**Figure 8.** An example of a bad result using the inpainting method.

Finally, removed regions from images are replaced using similar regions on other images, using an inpainting method. As we work with image sequences, temporal redundancy is used in order to find similar scenes for image inpainting.

Although achieved results are promising, we found some problems finding images to be used as source patches on the inpainting process, obtaining bad reconstruction of the image. This problem can be solved using an extended set of images or, since we have georeferenced images, using spatial redundancy in order to find images captured near the analyzed scene.

## Acknowledgements

This work was partially supported by two research grants: TIN2006-15308-C02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

## References

- [1] J. Casacuberta, J. Miranda, M. Pla, S. Sanchez, A. Serra, and J. Talaya, "On the accuracy and performance of the geomobil system," in *International Society for Photogrammetry and Remote Sensing*, 2004.
- [2] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 39–51, 1998.
- [3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on CVPR*, vol. 1, 2001, pp. I–511–I–518.
- [4] A. A. M. Z. B. W. a. B. H. A. H. Andrea Frome, German Cheung and L. Vincent, "Large-scale privacy protection in google street view," in *IEEE ICIP 2002*, 2009, pp. 900–903.
- [5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.
- [6] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *IEEE ICIP 2002*, 2002, pp. 900–903.
- [7] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.
- [8] W. Jiang, "Human feature extraction in VS image using HOG algorithm," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*. IEEE Computer Society, 2007, pp. 886–893.
- [9] J. Hays and A. Efros, "Scene completion using millions of photographs," in *Proceedings of the 2001 IEEE Computer Society Conference on CVPR*, vol. 1, 2007, pp. I–511–I–518.



# Bibliografia

- [1] J. Casacuberta, J. Miranda, M. Pla, S. Sanchez, A. Serra, and J. Talaya, "On the accuracy and performance of the geomobil system," in *International Society for Photogrammetry and Remote Sensing*, 2004.
- [2] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, Hartmut Neve, and L. Vincent, "Large-scale privacy protection in google street view," in *IEEE ICIP 2009*, 2009, pp. 900–903.
- [3] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 39–51, 1998.
- [4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on CVPR*, vol. 1, 2001, pp. I–511–I–518.
- [5] D. Gernimo, A. M. López, A. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driven assistance systems." *TPAMI*, 2009.
- [6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.
- [7] S. Escalera, X. Baró, J. Vitrià, and P. Radeva, "Text detection in urban scenes," in *Frontiers in Artificial Intelligence and Applications; Vol.202*, 2009, pp. 35–44.
- [8] R. Alamus, A. Baron, J. Casacuberta, M. Pla, S. Sánchez, A. Serra, and J. Talaya, "Geomobil: a land based mobile mapping system for cartographic data capture proceedings of the xxii international cartographic conference of the ica." A Coruña, Spain, 2005.

- [9] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: a statistical view of boosting,” Dept. of Statistics, Stanford University Technical Report, 1998.
- [10] J. Hays and A. Efros, “Scene completion using millions of photographs,” in *Proceedings of the 2001 IEEE Computer Society Conference on CVPR*, vol. 1, 2007, pp. I-511–I-518.